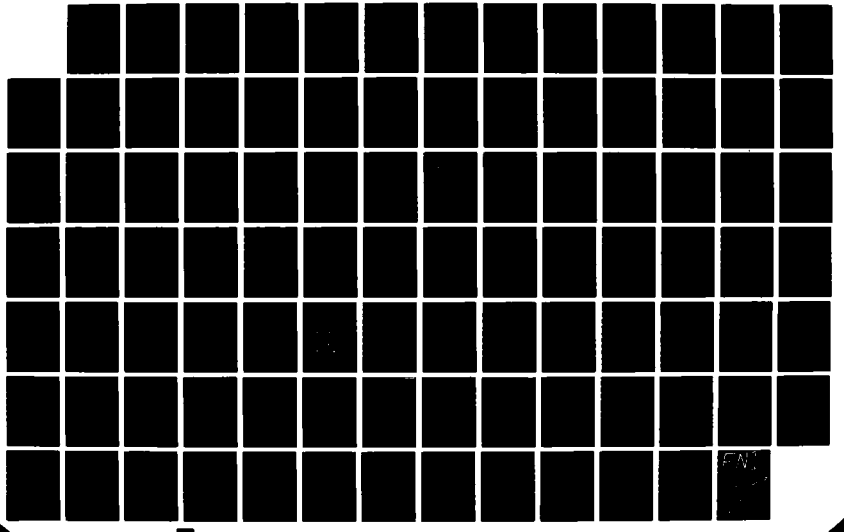
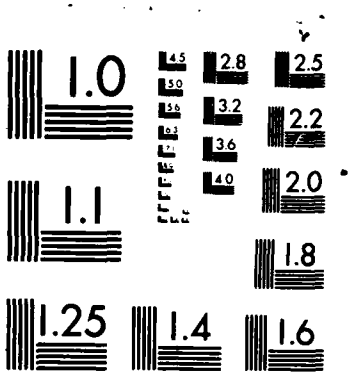
AD-A182 511 OPTICAL SYMBOLIC COMPUTING(U) COLORADO UNIV AT BOULDER 1/1
DEPT OF ELECTRICAL AND COMPUTER ENGINEERING
W T CATHEY ET AL. 30 APR 87 153-6920 AFOSR-TR-87-0751
UNCLASSIFIED AFOSR-86-0189 F/G 12/9 NL





SECURITY CLASSIFICATION OF THIS PAGE

2

EPORT DOCUMENTATION PAGE

AD-A182 511	1b. RESTRICTIVE MARKINGS
.E	3. DISTRIBUTION/AVAILABILITY OF REPORT unlimited DTIC FILE COPY
4. PERFORMING ORGANIZATION REPORT NUMBER(S) 153-6920	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 87-0751
6a. NAME OF PERFORMING ORGANIZATION Dept. of Elect. & Computer Engineering	6b. OFFICE SYMBOL (If applicable)
7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research	
6c. ADDRESS (City, State and ZIP Code) Campus Box 425 Boulder, Colorado 80309-0425	7b. ADDRESS (City, State and ZIP Code) Bldg. 410 Bolling AFB, D.C. 20332
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NE
9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-86-0189	
8c. ADDRESS (City, State and ZIP Code) Same as 7b	10. SOURCE OF FUNDING NOS PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2305; TASK NO. B1; WORK UNIT NO.
11. TITLE (Include Security Classification) Optical Symbolic Computing (U)	
12. PERSONAL AUTHOR(S) Cathey, Wade Thomas and Schmidt, Rodney A., and Moddel, Garret	
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 86-5-1 TO 87-4-30
14. DATE OF REPORT (Yr., Mo., Day) 87-4-30	15. PAGE COUNT
16. SUPPLEMENTARY NOTATION	
17. COSATI CODES FIELD / GROUP / SUB. GR.	18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Optical Computing, Artificial Intelligence, Symbolic Logic, Optical Processing.

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Computer simulation programs have been written to simulate optical symbolic systems. The simulation includes ferroelectric liquid crystals, detectors, and conventional optical components. Individual components have been assembled and tested. The next phase of the work is to assemble a simple, but complete, demonstration system, compare the results with the simulations, and to correct the simulations. The design and simulation of more complex systems will be performed.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21. ABSTRACT SECURITY CLASSIFICATION None
22a. NAME OF RESPONSIBLE INDIVIDUAL Dr Giles	22b. TELEPHONE NUMBER (Include Area Code) 767-4933
	22c. OFFICE SYMBOL NE

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE

OPTICAL SYMBOLIC COMPUTING

W. Thomas Cathey
University of Colorado
Boulder, Colorado 80309-0425

Several papers are attached that report on the work of the past year.

"Optical Representations for Artificial Intelligence Problems," was presented at the January 1986 SPIE meeting on Optical Computing, SPIE Vol. 625 (1986), pp 226-233 and a revised version is to be published in Optical Engineering.

"Optical Implementations of Mathematical Resolution," is to be published in the May issue of Applied Optics.

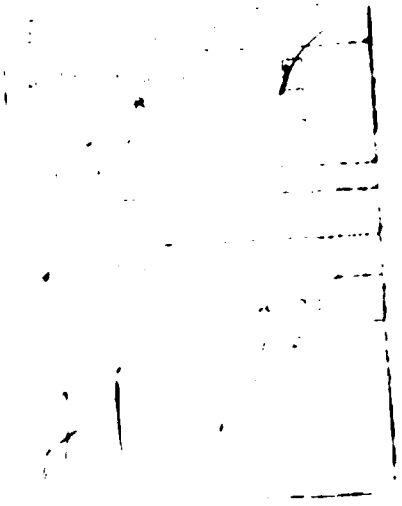
"Matrix-Vector Multiplication Using Polarization Rotators," was presented at the Optical Society of America's Topical Meeting on Optical Computing, Incline Village, Nevada, 16-18 March, 1987.

"Polarization-Based Optical Parallel Logic Gates Using Ferroelectric Liquid Crystal Spatial Light Modulators," was presented at the Optical Society of America Topical Meeting on Optical Computing, Incline Village, Nevada, 16-18 March, 1987.

"Photoaddressing of High Speed Liquid Crystal Spatial Light Modulators," was presented at the SPIE meeting on Optoelectronics, 13-15 January, 1987, Los Angeles, California.

"Low Loss Polarization - Based Optical Logic Gates," is to be presented at the International Commission for Optics, 24-28 August, 1987, Quebec.

"Polarization-Based Optical Parallel Logic Gate Utilizing Ferroelectric Liquid Crystals," is to be published in Optics Letters.



Optical representations

for

Artificial intelligence problems

Rodney A. Schmidt

University of Colorado at Denver

Department of Electrical Engineering and Computer Science

1100 14th Street, Denver, Colorado, 80202

and

W. Thomas Cathey

University of Colorado at Boulder

Department of Electrical and Computer Engineering

Boulder, Colorado 80309-0425

Abstract

Artificial intelligence problems are solved on electronic computers by techniques that make heavy use of address calculation and dynamic management of data storage space. Optical computing, on the other hand, is normally associated with numerical problems in which the size of the data space is fixed and addressing may be handled in a predictable manner not affected by actual data values. A criterion is presented for determining the amount of dynamic storage management required for an expert system problem and several methods are discussed for eliminating unnecessary address manipulation by careful choice of data representation. Major emphasis is placed on the implementation of the mathematical technique of resolution. Various resolution strategies are analyzed and the impact of these strategies on storage management is assessed with a view to minimizing the complexity of processing. Finally, novel uses of electro-optical/electronic hybrids are considered for problems in which the state space grows drastically or where reversible control strategies are required to implement search methods.

Subject terms: digital optical computing; optical logic; artificial intelligence; expert systems; mathematical resolution.

Contents

1. Introduction

The speed and parallelism inherent in optical computing has made it an attractive technology for a variety of computationally intensive problems. Optical computing has been successfully applied to signal processing[1], equation solving[2], and digital logic[3], among other fields. However, in artificial intelligence, one of the most computationally intense fields, only very limited progress has been made. A breakthrough in this area would have tremendous benefits, particularly to such programs as the DARPA Strategic Computing Initiative.

Progress is slow in A.I. because it is qualitatively

different from the disciplines in which optical computing is succeeding. It is the intent of this paper to discuss these differences. Some of them are fundamental and must be accomodated by any optical AI architecture. Others are historical, and are the product of methods designed for implementation on sequential electronic processors. These can be sidestepped by proper choice of representation and processing strategy. This paper will present some examples.

2. History and properties of symbolic computing

Unquestionably, the most profound early influence on artificial intelligence (at least on the subset called symbolic computing) was the development of the LISP programming language by John McCarthy at MIT in the late 1950's[4,5]. For almost 30 years, LISP has been the "lingua franca" of symbolic computing and artificial intelligence. The approaches to most AI problems are so commonly expressed in LISP that it is difficult to imagine any other way to deal with them. This mindset must be surmounted if new technologies such as parallel processing and optical computing are to be applied to AI problems.

LISP is a language intended originally for serial interpretive execution. As such, it has several major characteristics which mesh well with this environment. First, the representation of programs is the same as the

representation of data. LISP is the only high level language in which a program can legitimately generate new instructions and execute them "on the fly". While this feature is little used, the technique of "procedural embedment", or mixing program fragments in a basically data-driven problem solution, is used in many AI codes. Newer generations of LISP are de-emphasizing this technique. This is fortunate, since optoelectronic implementation of this feature would be very difficult.

A second property of LISP is a very high level of data abstraction. Storage management in a LISP program is handled entirely by the interpreter or runtime system. The programmer makes no commitment to the size or structure of the data objects that he manipulates and the locations and storage relationships of the data entities are unknown to him. This is a very positive feature to the programmer, who no longer has to dimension arrays, worry about runtime stacks or manage data in a highly recursive, dynamic execution environment. The price that is paid is that a substantial fraction of the memory space contains addresses, not data, and that unknown to the programmer, most of the computation revolves around address manipulation rather than data examination. Indeed, the fundamental building block in LISP is the "cons cell", which contains no data and two addresses. Data-containing cells are a special case. This dependency on addresses has even less desirable ramifications for optical computing, as

we shall see later.

The third significant property of LISP is its tendency toward recursive rather than iterative solutions to problems. This is again attractive from a programming point of view, since most AI problems can be decomposed into subproblems structurally identical to the original with less complex data. Recursion is the natural way to solve these problems. Unfortunately, the implementation of recursion requires the saving and restoring of a great deal of control state in a processor. Sequential electronic processors have no problem with this, but as we shall see, parallel and optical processors are not well suited to the management of large amounts of control state.

In summary then, our problem in using optical computing for AI is not merely to transfer algorithms from one technology to another, but rather to redevelop concepts into algorithms in a different technological framework in which old techniques may in fact be counterproductive. It should not be surprising that this is a difficult task.

3. Properties of optical computing

One way of conceptualizing algorithms is to view them as consisting of data manipulation and control. Data manipulation is the process of combining or rearranging data values to obtain newer and more useful data values. Boolean logic, arithmetic, and rearrangement are all

examples of data manipulation. Control includes the selection of alternative manipulation and the repetition of manipulation based on data-dependent conditions. It also includes the management of the processor state required for the decomposition of problems into subprograms.

It is in the area of data manipulation that most optical computing work has been done. If we may call data manipulation "calculation", then the field of optical computing at present is heavily dominated by optical calculation. Control in optical systems requires optoelectronic or optoacoustic interfaces and the use of spatial light modulators to effect selection. Since the effective use of repetition requires optical gain and state variables, the use of repetition is limited in most systems and is replaced by non data dependent replication (cascading) of hardware components.

A glaring difference between electronic and optical computers is that in an electronic system, all components of an algorithm are stored in a uniform way (bit patterns in memory) and are somewhat interconvertible. By contrast, in an optical system, the "program" is hardware, addresses are geometrical or time coordinates, while data values are optically expressed. Thus taking data dependent action requires light patterns to modify light flow through hardware, while address calculation requires light patterns to affect system geometry - either coordinates or path lengths. The tools available to do this (spatial light

modulators) are extremely slow compared to data manipulation and propagation rates. This further inhibits the transfer of algorithms from sequential electronic processors, where control is fast and data manipulation is slow.

Our task while solving AI problems must therefore be to eliminate extraneous control interactions and cast the problems as completely as possible as calculation problems. The necesary residue of control must be accomodated by an optical system, but must be handled in a way that minimizes its impact on performance.

4. Data in symbolic optical computing

Symbolic computing is often described as "object-oriented". What is meant by this is that the presence or absence of a particular data object in a certain context is the most relevant fact. The value of the data object is much less important - indeed, the object may not even have a value. In conventional symbolic computing, the objects present in a certain context are grouped together in a list, which may be sequentially searched to determine presence or absence of specific objects. A list is an inconvenient representation for optical computing because of the nonequivalence of data and addresses. A more natural representation is an array of booleans, with each possible object assigned a fixed

position. In many AI problems, the universe of possible data objects is constrained to on the order of 1000 or less, so this is a feasible approach. The correspondence between a linked list and a boolean array is shown in Figure 1. This representation presumes that the order of elements in the list is not significant.

This simple representation does not allow for objects with a value. An important special case of valued objects occurs in the mathematical technique of proof by resolution. This powerful technique for validating or invalidating hypotheses requires boolean valued objects. These objects are arranged in clauses, which are disjuncts of selected boolean objects. Thus to represent clauses as boolean arrays requires more than one bit per data object (a data object is called a "literal" in resolution). Using two bits per literal allows the representation of the four states shown in Figure 2. The need for three of these states is obvious; the fourth state is a byproduct of the resolution process, and is discussed in detail later. The particular notation used in Figure 2 is called "double rail" or "differential" logic and simplifies some logical operations in resolution. Representations of three simple clauses are shown in Figure 3 using this notation. In this discussion, we assume that the discrete data values are represented by incoherent intensity values, although there are numerous other possibilities. In an actual implementation, the choice would depend on the current

state of the art in optical device technology.

One additional representational choice remains to be made. Since our system has both a space and a time domain, we may express our boolean vector in either of these dimensions. If we choose the space domain, all literals in a clause will occur in the same image. As we shall see later, this eliminates the need for control state variables during resolution, but limits us to one spatial dimension for the resolution itself. If a clause is expressed in the time domain, all literals in the clause occur at the same spatial location, but at different times (in different images). This allows two spatial dimensions for the resolution process, but requires two boolean control variables per resolved clause to remember the outcome of the process. If growth of the data space is accomodated in the time domain, mapping clauses into the time domain would appear less complex than the spatial alternative, but the decision will ultimately revolve around the availability of suitable optical memory for the control state.

One characteristic of AI problems which must be accomodated by optical computing representations is that the data space is dynamic. Unlike numerical problems which have a fixed number of variables whose values change, AI problems generate additional contexts and groupings of objects as the problem is being solved. There are basically two dimensions in which growth can occur: time and space. Growth in time implies that when images

interact during data manipulation, new images are created which supplement, not replace, the originals. These new images must be time sequenced in an optical pipeline with pre-existing images to define the new state of the computation. This requires that the pipeline get longer as computation proceeds. Alternatively, the new images can be spatially adjoined to the old ones, and the amount of area devoted to the image may increase. Unfortunately, the amount of potential growth in the data space is so large that this approach may often lead to exceeding hardware limitations.

AI problems (such as game playing) in which each decision requires both generation of a new state space and retention of the old are particularly susceptible to data space explosion. These problems use a reversible control strategy which must allow backup to earlier states of the solution while exploring for an answer. Resolution is a more attractive first problem for optical implementation because it uses irreversible control strategies. These do not have as pronounced a data growth.

AI problems can be categorized on a control/data space map as shown in Figure 4. Conventional numerical problems are shown in this figure for comparison. The data complexity axis can be classified into four regions according to the type of data manipulation:

Substitution: the state space is fixed, and

computation substitutes new values for old.

Replacement: computation generates a new state space which replaces the old.

Augmentation: computation generates new elements to add to the existing state space.

Replication: computation creates a new state space but all old state spaces must be retained.

Problems toward the left side of the map are most amenable to optical implementation. This includes some, but not all, forms of resolution.

5. The process of resolution and its optical implementation

Resolution is a mechanical process for detecting contradiction within a family of assertions. It may be used for proof of an assertion by showing that the negation of the assertion contradicts the other assertions. A detailed discussion of resolution is beyond the scope of this paper, but several excellent references exist[6,7]. The process may be summarized as follows:

(1) The original set of axioms (rules) are converted into clause form. In clause form, logical implication

is removed from the rules using the equivalence of the forms (P --> Q) and (P' + Q). The rules are placed in a canonical disjunctive form. Figure 5 illustrates some simple rules transformed into clauses. This process need only be performed once for a given problem domain. All clauses are assumed to be true.

(2) The assertion to be proven is negated and converted to clause form.

(3) Pairs of clauses are selected and their resolvent formed. A pair of clauses can be resolved when exactly one literal occurs in both clauses, asserted in one and denied in the other. This is an application of the logical tautology: (A+B)(A'+C) --> (B+C). The resolvent is the union of the source clauses with the single contradicting literal removed.

(4) Step 3 is repeated using both the original assertions and new resolvents until (a) no new resolvents can be formed; or (b) a resolvent is formed with no literals at all. It is also possible that the computation is nonterminating. Case (b) is the desired case, and shows that the original assertion was true.

The key to resolution lies in how the pairs of clauses

are selected in step 3. A number of different methods exist. Central to all of them is that most pairs of clauses cannot be resolved because they are not in the required form. Thus an intermediate step is to determine whether or not a particular pair of clauses can be resolved. Clearly, the answer for any pair of clauses is independent of the answer for any other pair. Thus all possible pairs of clauses can be checked in parallel. This process is illustrated in Figure 6 for time serial clauses and in Figure 7 for spatially parallel clauses. This has been discussed in more detail elsewhere [8]. In both cases, a light pattern is produced for each literal which may be decoded into the four double rail values discussed earlier. This is shown in Figure 8.

A pair of clauses may be resolved if exactly one literal in the vector is in the contradiction state. The resolvent has in fact already been formed by the checking process, except that the contradicted literal must be output in the "not present" state in order to participate in further resolution. This may be accomplished using a spatial light modulator to remove the literal from the output.

A major problem in resolution is that only a small fraction of potential resolvents meet the single contradiction requirement. Thus after a vector or matrix of potential resolvents is formed optically, only a few of the results are useful. Many passes through the resolvent

forming mechanism are required in order to solve a problem, and if the useless clauses are not discarded, the growth of the data space will be exponential with a very large base. We have discussed how to determine that a particular clause is useless - our problem now is how to eliminate the useless clauses. Functionally, we need to arrange the result matrix to squeeze out the useless clauses, then adjoin the remaining elements with the original clause matrix to form the next data set for input to the resolution mechanism. With present technology, this will require another spatial light modulator or electro-optic conversion.

The approach to resolvent compaction will vary depending on whether time-serial or spatially parallel notation is used. In time serial notation, the one-contradiction condition is detected by the use of a pair of control variables C (contradiction) and X (excess contradictions) calculated by an inductive formula. If

$$C\langle 0\rangle = 0 \qquad (1)$$

$$X\langle 0\rangle = 0$$

and

$$C\langle n\rangle = (T\langle n\rangle)*(F\langle n\rangle) + (C\langle n-1\rangle) \qquad (2)$$

$$X\langle n\rangle = (T\langle n\rangle)*(F\langle n\rangle)*(C\langle n-1\rangle) + (X\langle n-1\rangle)$$

then a resolvent is acceptable iff $C\langle final\rangle = 1$ and $X\langle final\rangle = 0$. T and F are the true and false components of the double-rail notation for each literal, and the

operations * and + are boolean conjunction and disjunction, respectively. Since X may become true as late as the passage of the last literal through the resolvent mechanism, the activation of the SLM may not begin until that time. Thus the SLM must be located some distance from the optical logic which forms the resolvent. In order to "squeeze out" unacceptable resolvents from the result, one SLM per result must be provided. This SLM will either perform no operation on the output images (if the associated resolvent is acceptable), or will move all remaining resolvents up one position (if the associated resolvent is unacceptable). This could be a rather large number of SLM's. A one dimensional example is shown in Figure 9. In this figure, each SLM squeezes out the corresponding clause if it is invalid. The bottom clause need never be squeezed out.

If spatially parallel notation is used, the acceptance criterion can be computed in parallel using a tree structure to capture the necessary contradiction condition. Denoting the level in the tree by the first index, we have

$$C\langle 1,i\rangle = (T\langle i\rangle)*(F\langle i\rangle) \tag{3}$$

$$X\langle 1,i\rangle = 0$$

Then for each succeeding level we have

$$C\langle n,i\rangle = (C\langle n-1,2i\rangle) + (C\langle n-1,2i+1\rangle) \tag{4}$$

$$X\langle n,i\rangle = (C\langle n-1,2i\rangle)*(C\langle n-1,2i+1\rangle) + (X\langle n-1,2i\rangle) + (X\langle n-1,2i+1\rangle)$$

The condition for success is that C<final,0> = 1 and X<final,0> = 0. This is the same condition as derived for the time-serial case, but no storage is required. As in any binary tree, the number of levels is $\log_2$ of the number of possible literals, perhaps 8-10 for reasonable cases. With spatially parallel notation, one dimension of the image is devoted to the literals themselves. The potential resolvents between a clause and an array of clauses are distributed across the other dimension. Thus while the same sort of compaction with SLM's is required as in the time-serial case, the compaction is uniform in one dimension. This will allow simpler SLM architecture. In particular, 2^k resolvents can be compacted using k SLM's as shown in Figure 10. The SLM's are arranged in order such that the first SLM can move a clause one position, while the kth SLM can move a clause by 2^{k-1} positions. Each succeeding SLM requires only half the independent elements as its predecessor. Control for the SLM can be electronically derived from the sequence of (C,X) bits for the various resolvents.

6. Comparison of resolution strategies

The remaining major issue in resolution is how the pairs of clauses to be resolved are selected. The most straightforward strategy is breadth-first resolution. In this strategy, the initial clauses are considered the "base

generation". Generation 1 is formed by attempting to resolve each member of the base generation with all other members of the base generation. Generation 2 is formed by resolving each member of generation 1 with all the members of generation 1 and the base generation. This process is repeated for each succeeding generation.

Breadth-first resolution is complete, in the sense that it will find a contradiction if one exists, but is very expensive in storage and computing time. A normal assumption in resolution is that the original axioms do not contain a contradiction; if one exists, it is generated by the negation of the theorem to be proved. Thus, generating resolvents between the original axioms does not directly advance the search for a contradiction. The assumption of consistency of the original axioms leads to the second strategy.

In set-of-support resolution, the base generation contains the original axioms, and generation 1 contains only the negated hypothesis. Resolution then proceeds in a breadth-first manner, but since generation 1 has only one clause, relatively few clauses are produced in generation 2. This approach produces more generations, but fewer clauses in each, and is usually more efficient due to the special significance of the negated hypothesis. Set-of-support resolution is also guaranteed to find a contradiction if one exists.

A subset of breadth-first resolution is linear-input

resolution. In this strategy, the next generation is produced by resolving the current generation only against the base generation. Thus one of the parents of each resolvent is static and fixed. Linear-input is an attractive and commonly used strategy, but is not complete. Its appeal for optical implementations is that the base generation could be captured in transmission masks, and only the generated resolvents need be stored in images. Further, since each generation of resolvents is replaced by a new generation, there is little or no storage growth. Although it is an incomplete strategy, linear-input resolution may be useful in problems with many facts (single literal clauses) and few implications (multiple literal clauses) since it is the resolution of single literal clauses which ultimately leads to final contradiction (null clauses).

An interesting strategy related to linear-input resolution is ancestry-filtered resolution. In this strategy, which is complete, one of the pair of clauses to be resolved must be an ancestor of the other clause. This offers a considerable reduction in the number of resolvents to be considered, but requires parentage information about clauses. In conventional systems, this is done with linked lists, but this is not optically feasible. We have not examined this strategy in detail, but it is possible that some amount of clause replication in redundant data spaces may substitute for list structure, and make this approach

feasible. It is clear that the control and data management problems of this strategy are the worst of the four strategies discussed.

7. Electro-optic hybrids

At the present time, it is difficult to see how a completely optical resolution system could be built. The primary areas of difficulty are optical gain in a system with a large number of bits, and in optically accomodating the growth of the state space. These problems could be gracefully solved, along with some of the control problems mentioned earlier, by the use of electro-optical conversion. It appears that a single type of device would suffice - an optically addressable memory. Such a device would consist of a 2-D array of cells, each containing a photodetector, one bit of electronic memory, and a light emittting diode (LED). Upon electronic command, the device would load the array of bits from an image, or generate such a image via the LED's. The memory could also be read or written electronically. Such a device could be used for gain, for data input and output, and for control, with suitable electronic interfaces. Portions of such devices already exist in CCD arrays and solid-state cameras. The introduction of LED's into such devices would be a logical step in the evolution of GaAs technology.

The use of an optically addressable memory in

resolution would eliminate much of the need for SLM's, since the compaction of resolvents mentioned earlier could be achieved electronically by rewriting the memory. Inclusion of shift register circuitry in the memory would even allow the compaction to be done in parallel.

With data stored electronically, expansion of data storage need not be accomodated in the optical portion of the system. Instead, bulk electronic memory could be substituted. Further, introduction of the initial resolution clauses into the system could use the same mechanism.

8. Summary

Our purpose in this paper has been to review the essentials of symbolic computing, demonstrate list-free notation for at least one problem of interest, and show the feasibility of optical techniques for a significant portion of this problem.

The problem of mathematical resolution was discussed from the standpoint of basic operators, control complexity and data management complexity. The basic operations required are clearly feasible with optical technology, while control and data management require either advances in technology or use of electronic hybrids. Our future work will involve quantifying the magnitude of these issues for actual problems and attempting to match specific

optical and electro-optical technologies to their solution.

9. Acknowledgements

Publication of this work was supported by a grant from the Air Force Office of Scientific Research. The underlying motivation came from a talk by John A. Neff, Defense Advanced Research Projects Agency.

10. References

1. B. Moslehi, J.W. Goodman, M. Tur, and H.J. Shaw, "Fiber-Optic Lattice Signal Processing", Proc. IEEE 72(7), 909 (1984).

2. S.H. Lee, "Optical Analog Solutions of Partial Differential and Integral Equations", Opt. Eng. 24(1), 041 (1985).

3. A.A. Sawchuk, and T.C. Strand, "Digital Optical Computing",Proc. IEEE 72(7), 758 (1984).

4. J. McCarthy, "Recursive Functions of Symbolic Expressions and Their Computation by Machine", Comm. ACM 3(4), 184 (1960).

5. P. H. Winston and B.K.P. Horn, LISP, Addison Wesley, Reading (1981).

6. N. Nilsson, Principles of Artificial Intelligence, Tioga, Palo Alto (1980).

7. P.H. Winston, Artificial Intelligence, Addison Wesley, Reading (1984).

8. R.A. Schmidt and W.T. Cathey, "Optical Implementations of Mathematical Resolution", Applied Optics, 26(8), nnn (1987).

Rodney A. Schmidt received his B.S.E.E from M.I.T. in 1966 and his M.S.E.E. and Ph.D. from Stanford University in 1971. Hw worked for ESL, Inc. in Sunnyvale, Ca. from 1971 to 1976. He was an assistant professor at the University of Denver from 1976 to 1978, and was Manager of Software Engineering at Denelcor from 1978 to 1982. Since 1982 he has been an assistant professor of computer science at the University of Colorado at Denver. His technical interests include artificial intelligence, parallel computing and robotics.

W. Thomas Cathey has been at the University of Colorado for 18 years, and is now Professor of Electrical Engineering and Director of the Center for Optoelectronic Computing Systems. Previously, he was at the Autonetics Research Center of Rockwell International.

Dr. Cathey's degrees are from the University of South Carolina and Yale University. His research has been in the fields of holography, adaptive optical arrays, imaging and optical information processing. Recently, his work has been in the areas of optical computing, imaging systems, and optical sensing systems.

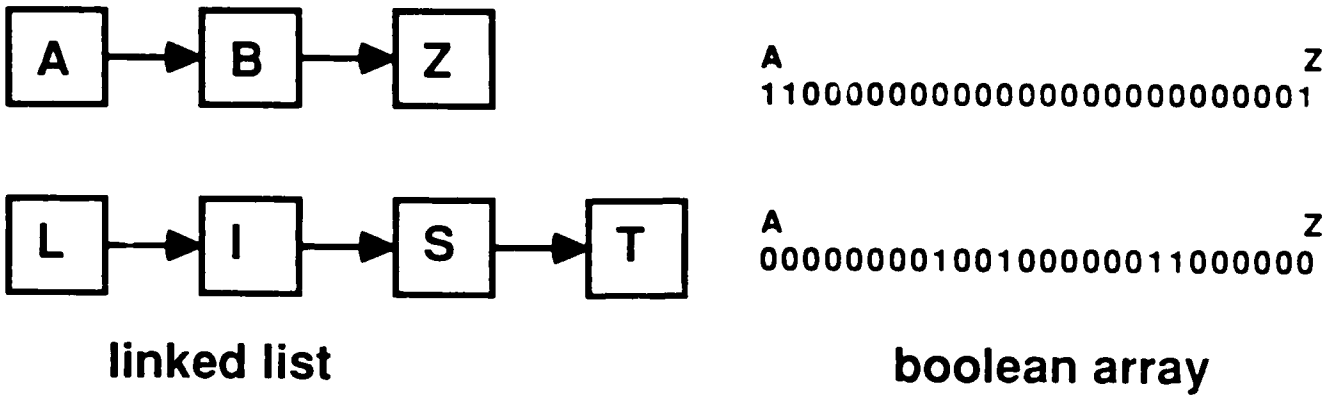


Figure 1 - Lists as Boolean Arrays

dibit		interpretation
00	-	not present
01	-	false (negated)
10	-	true (asserted)
11	-	contradiction (both asserted and negated)

Figure 2 - Differential Boolean Logic

	A			D
A + B' + D	10	01	00	10
B + D	00	10	00	10
A + D	10	00	00	10

Figure 3 - Clauses in Differential Form

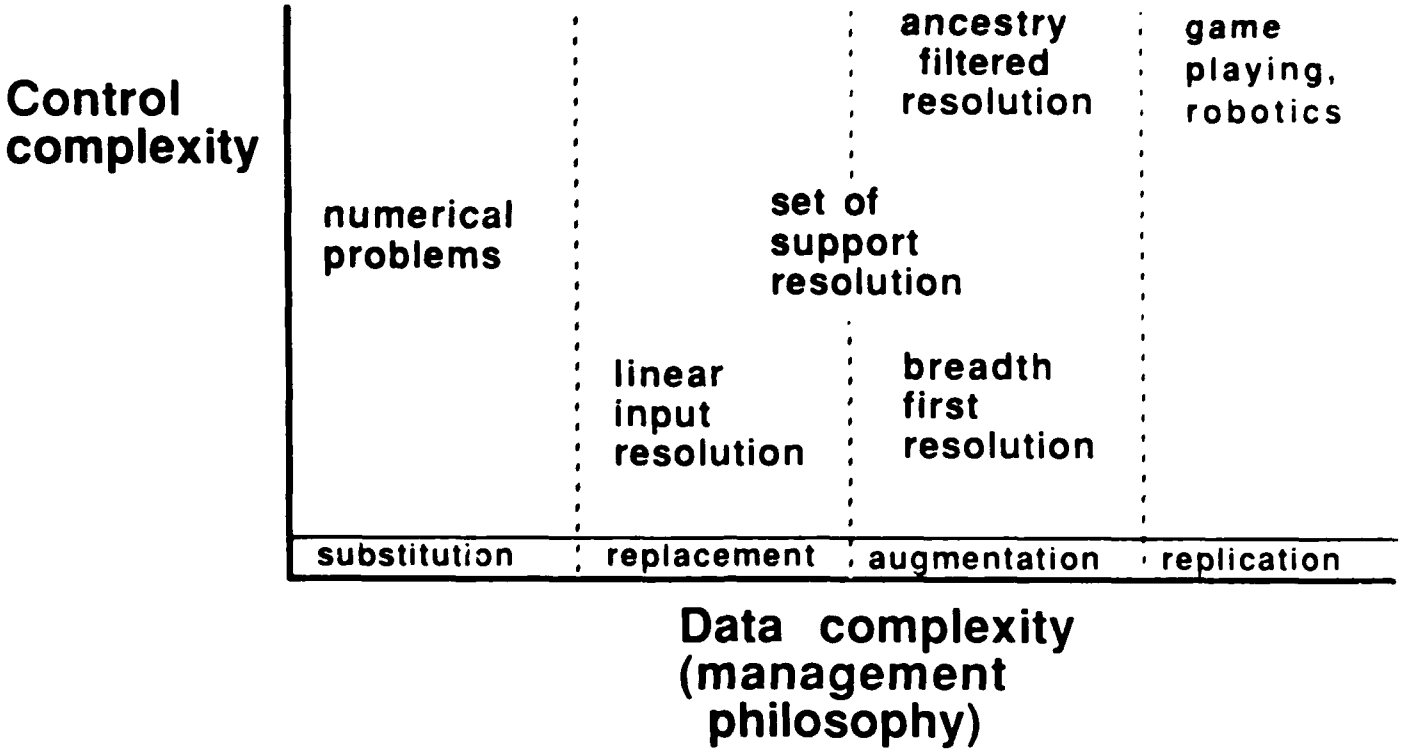


Figure 4 - Control and Data Complexity of Algorithms

inference	clause form
A'B → D	A + B' + D
B' → D	B + D
A → D	A' + D
A + B → D	A' + C B' + C
A → CD	A' + C A' + D

Figure 5 - Inference Rules in Clause Form

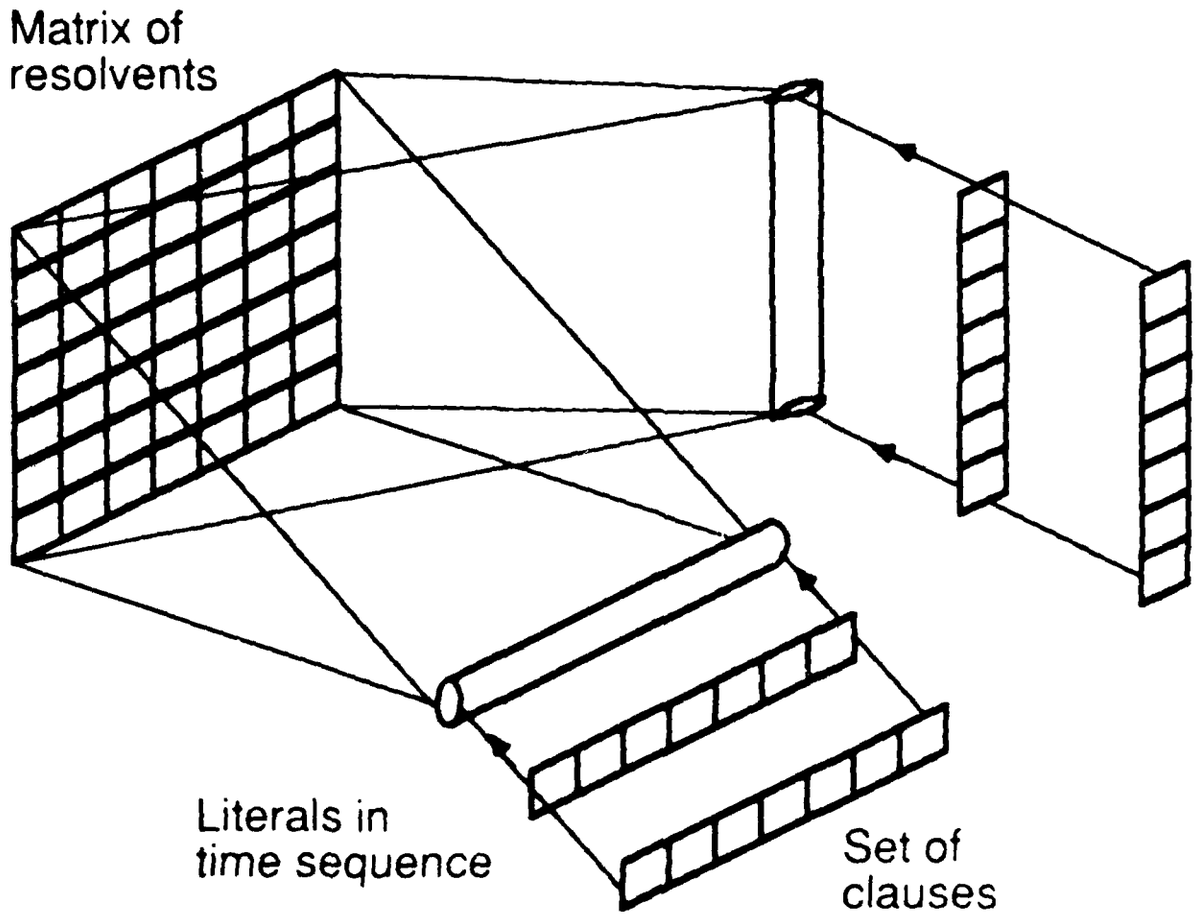


Figure 6 - Time Serial Resolution

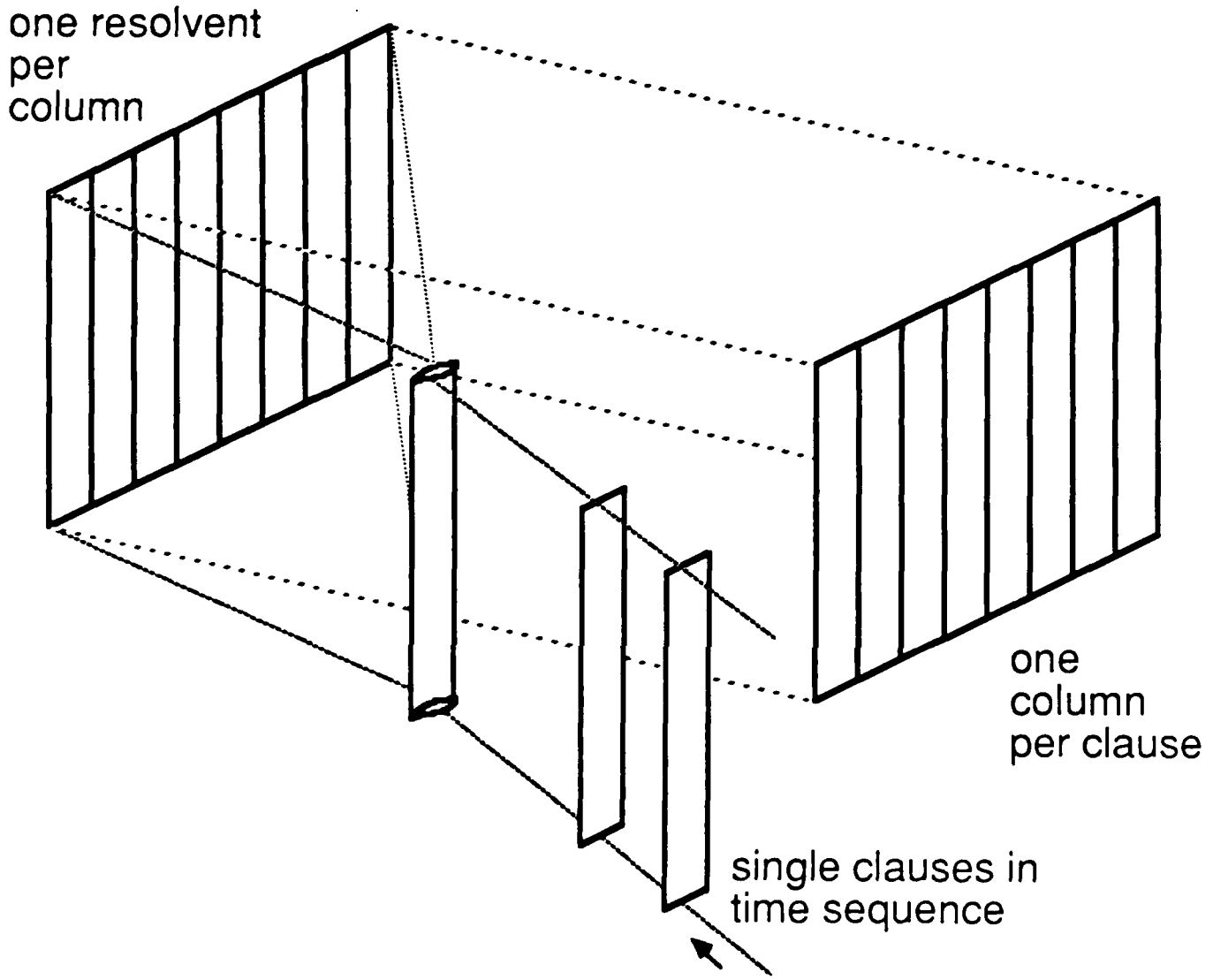


Figure 7 - Spatially Parallel Resolution

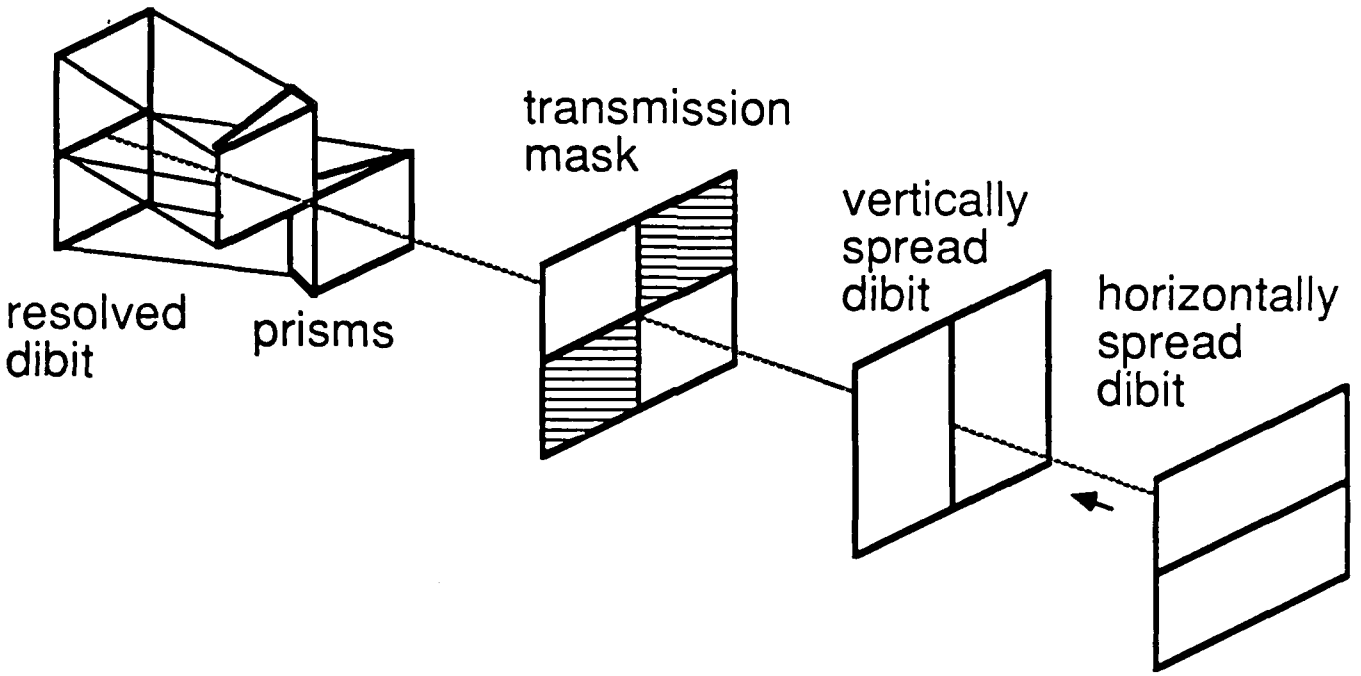


Figure 8 - Time Serial Literal Formation

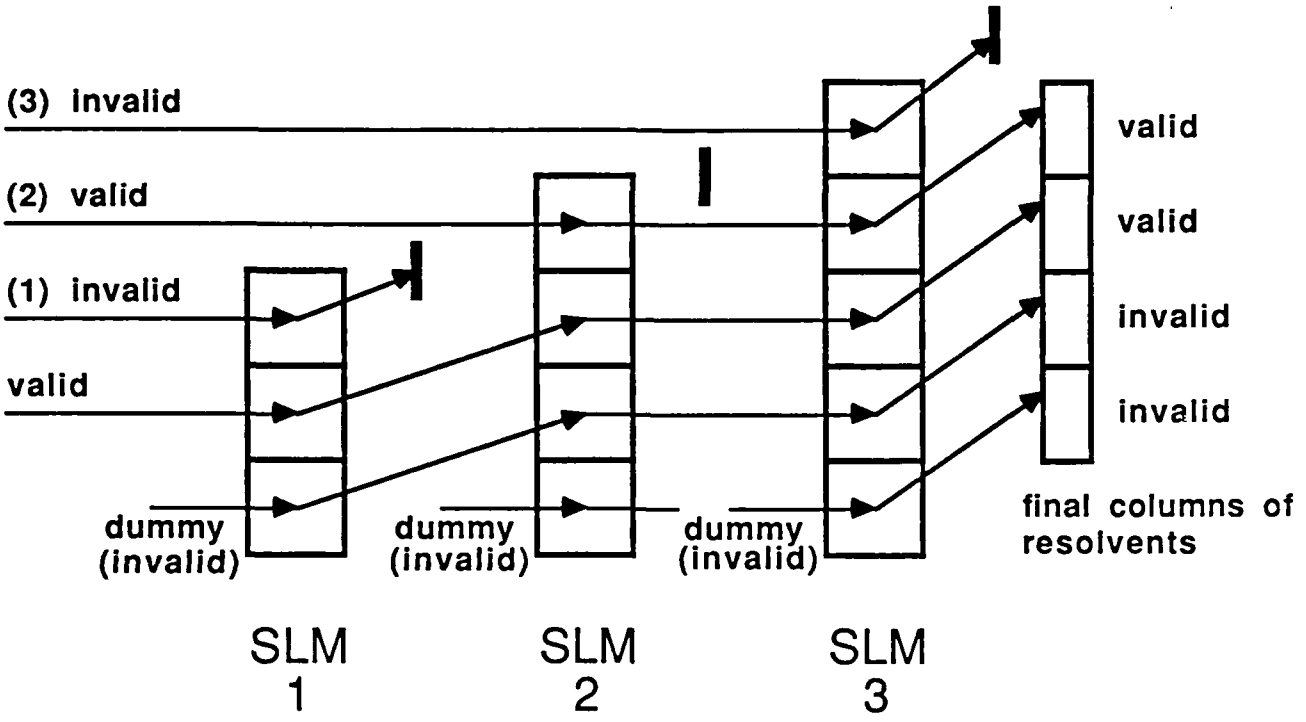


Figure 9 - Time Serial Resolvent Compaction

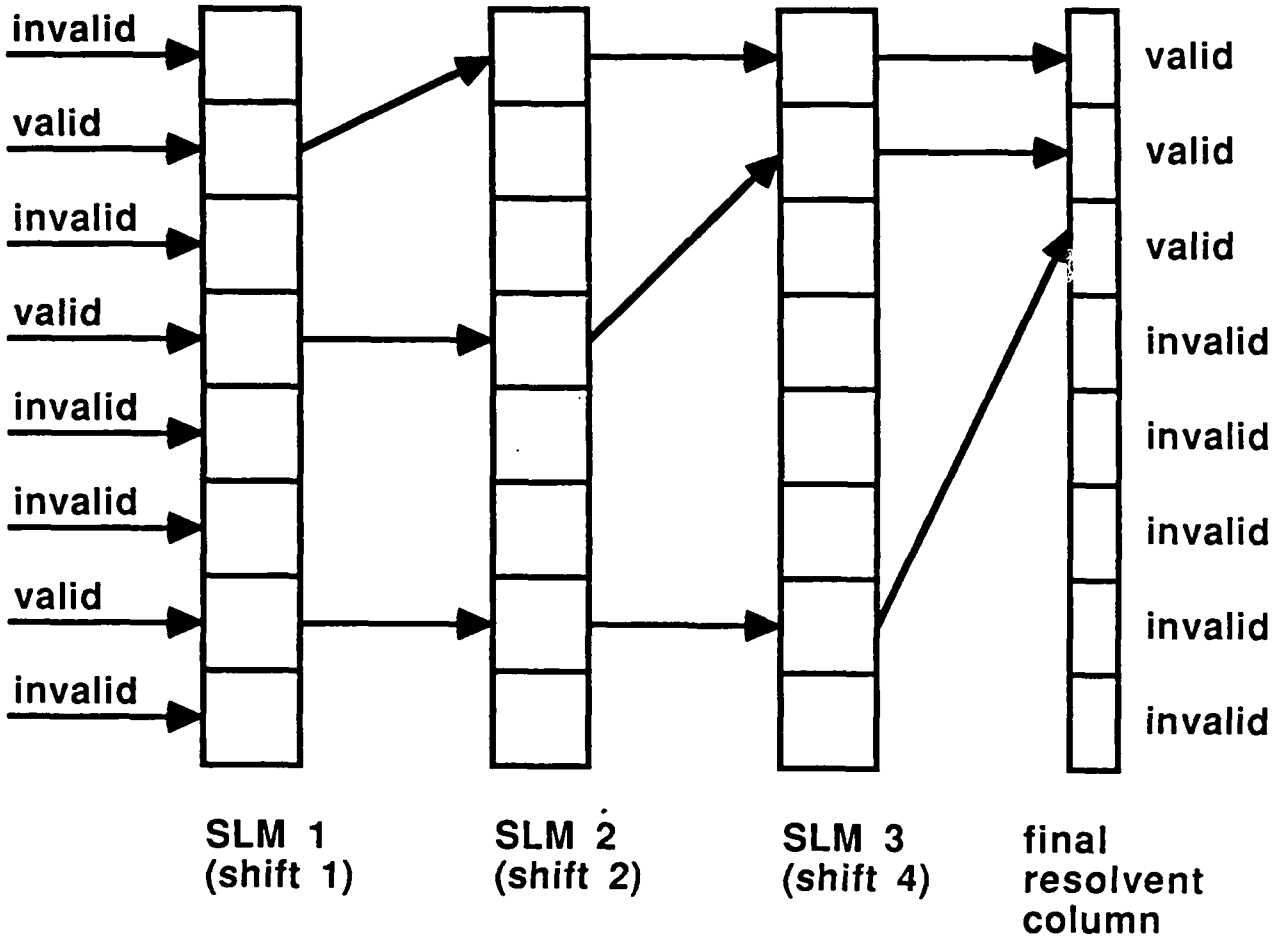


Figure 10 - Parallel Resolvent Compaction

OPTICAL IMPLEMENTATIONS

OF

MATHEMATICAL RESOLUTION

Rodney A. Schmidt
University of Colorado, Denver
Department of Electrical Engineering and Computer Science
1100 14th Street
Denver, Colorado 80202

W. Thomas Cathey
University of Colorado, Boulder
Electrical and Computer Engineering Department
Boulder, Colorado 80309-0425

Mathematical resolution is an algorithmic technique for reasoning from facts expressed in clause form to a conclusion. The technique is normally implemented on electronic computers with list-processing languages. This paper presents data representation and processing techniques for a parallel implementation using array-based optical logic. Implementations up through the quantified propositional calculus are presented, and the operations of resolvent formation, unification and search are discussed. It is shown that a largely parallel formulation of resolution is possible, and optical technologies are suggested to implement this formulation.

INTRODUCTION

Optical computing is an attractive technology for a variety of problems due to its speed and parallelism. It has been successfully applied to signal processing, equation solving, and digital logic, as well as to other fields [1-4]. The application of optical computing to artificial intelligence (A. I.) has been limited, due to the qualitatively different nature of A. I. problems. Areas in which optical computing has been successful are ones in which the control of algorithms is not data-dependent. A. I. is at the opposite end of the computational spectrum. In A. I., the flow of control through an algorithm is dominated by the data, especially in conventional A. I. languages such as LISP. This creates problems for optical implementations, because optically controlled switching of optical data typically requires opto-electronic data conversion and uses relatively slow spatial light modulators. A practical optical implementation must minimize these interactions.

We focus here on a particular subset of A. I., namely expert systems, and on a particular (albeit general) method within the subset. The method is mathematical resolution, a method for deductive reasoning about facts and hypotheses. Mathematical resolution is an algorithmic technique for mechanizing the deductive reasoning process so that it can be mechanically carried out by a computer. The technique is extremely computer-intensive, and also generates large amounts of intermediate data during the derivation of a proof. Many aspects of the process are inherently parallel, and this fact makes optical implementations attractive. An unattractive feature is the requirement to manage the data space whose size changes in a data-dependent fashion. The

challenge is to design a data representation and a set of operations which implement mathematical resolution with as little interaction as possible between control and data, and then implement the design using optical components.

MATHEMATICAL RESOLUTION

Mathematical resolution uses boolean logic to generate new statements through deductive reasoning from an initial statements set. Normally, a hypothesis is provided by showing that its negation contradicts the original fact set. In resolution, a statement or rule of inference is represented in a special form called a *clause*. A clause is a disjunction of terms, each called a *literal*. Since in ordinary logic, rules are represented by logical implication, a preprocessing step is required to transform ordinary statements and rules into a form suitable for resolution. This process need only be performed once for a given problem area and is not covered here. Several excellent references exist [5-7]. As an example of a transformed rule, the implication $(P \rightarrow G)$ transforms by the rules of Boolean algebra into the clause $(P' + G)$.

The next process in mathematical resolution is resolvent formation. In this process, new statements are derived from sets of old statements. For example, the set of clauses $\{(P' + G)\ ;\ G'\}$ resolve to generate the new statement $\{P'\}$. The new statements are added to the rule base and resolvent generation continues until the null clause is generated or no new clauses are possible. If the null clause is generated, the original hypothesis was true, since its negation produced a contradiction.

In practical applications of resolution, hundreds or thousands of resolvents must be generated before a problem is solved. The order in which the resolvents are generated may have a dramatic effect on how many resolvents are required and how long the process takes. At the same time, some orderings require extremely sophisticated algorithms for control, while others are relatively straightforward. A tradeoff must be made between efficiency and complexity, especially for optical implementations where complex control is very difficult.

The technique of mathematical resolution can be applied to several different levels of mathematical logic. Each higher level adds complexity to the process but allows the representation of more interesting reasoning problems. Optical implementation of the lowest level seems straightforward, but restricts the problem domain to rather trivial problems. Implementation of the highest level poses severe design problems which at present do not appear to have easy solutions. Our expectation is that at an intermediate level interesting problem domains can be handled without taking present optical methodologies beyond their breaking points.

The lowest level of mathematical logic considered is the *propositional calculus*. In the propositional calculus, the literals in clauses are Boolean constants, in either asserted or negated form. The clause example given earlier was drawn from the propositional calculus. Mathematical resolution in the propositional calculus is straightforward. Two clauses resolve if and only if exactly one literal occurs in asserted form in one of the clauses and in negated form in the other clause. Thus $\{(P' + Q)\ ;\ P\}$ resolve to $\{Q\}$. If more than one literal meets the test, then the clauses do not resolve. Thus

$\{(P' + Q) ; (P + Q')\}$ do not resolve. This latter case makes sense, since the two clauses represent $(P \rightarrow Q)$ and $(Q \rightarrow P)$, from which no new facts can be deduced. If zero literals meet the test, then likewise no resolvent exists. Thus $\{(P' + Q) ; P'\}$ do not resolve, since no literal occurs in one clause asserted and in the other negated. This corresponds to the logical situation that $(P \rightarrow Q)$ and (P') do not permit further deduction.

The next level of mathematical logic is called the *quantified propositional calculus*. In the quantified propositional calculus, the literals in a clause are Boolean valued functions, called *predicates*. Each predicate has zero or more arguments which may be constants or variables. This formulation allows the specifications of general rules which relate different predicates of variables. The variables themselves are not Boolean valued, nor are the constants. The domain of the predicate functions may be any discrete or continuous number space. A sample clause in the quantified propositional calculus might be $(P'(x) + Q(x))$. In this clause, P and Q are predicates, and the variable x is the argument for each predicate. Resolvent formation in the quantified propositional calculus involves an additional step. A set of clauses can be resolved if exactly on literal is asserted in one clause and negated in another *after the process called unification is performed.*

Unification involves the trial substitution of constants for variables, or variables for variables in an attempt to make two clauses more similar. If a substitution can be found which makes two clauses meet the basic resolution requirement, then a resolvent can be formed. For a given pair of clauses, several such substitutions may exist, each yielding a different resolvent. A sample unification and resolution might be $\{(P'(x) + Q(x)) ; Q'(A)\}$, where

A is a constant. This pair resolves to $P'(A)$. The unifying substitution is $x \rightarrow A$, yielding the clause pair $\{(P'(A) + Q(A)) : Q'(A)\}$. These meet the normal resolution condition, and generate the resolvent shown above.

Resolution in the quantified propositional calculus is substantially more complex than in the propositional calculus. Not only do various unifying substitutions have to be explored, but predicates can occur more than once in a clause, each time with different arguments. Further, while the order of predicates in a clause is not significant, the order of arguments to predicate functions very definitely is significant. All of these issues lead to more complex representations and operations.

The most general form of mathematical logic used in resolution is the *first order predicate calculus*. This notation allows not only constants, predicates and variables, but also arbitrary functions as arguments of predicates. These functions allow real-world relationships between variables to be modeled. The potential complexity of handling these functions optically is enormous, and the effort reported here does not address the first order predicate calculus. The major consequence of ignoring the first order predicate calculus is to either limit the size of problems which can be handled, or to drastically increase the number of clauses required to represent the problem in the quantified propositional calculus.

In any of the logic schemas described above, the proof process is interactive. In most cases, the resolvents directly derivable from the original facts and rules (the base set), do not themselves solve the problem. The resolution process must be repeated using the new resolvents to produce yet another generation of resolvents. This iteration of resolvent generations may be repeated

dozens of times before the original hypothesis is proved. There are, however, several alternatives in how the next generation of resolvents is produced. Several of these appear to have potential for optical implementation.

In *breadth-first* resolution, generation 0 (the base set) contains all the original facts and rules, including the negation of the hypothesis to be proved. Generation 1 contains all the clauses directly deducible from generation 0. Generation k is produced by resolving generation k-1 against the union of generations 0, 1 . . . k-1. This strategy will exhaustively produce all possible resolvents, and will prove the hypothesis if it is true. The quantity of resolvents generated is large and may exceed any reasonable implementation limits.

In *linear-input-form* resolution, generations 0 and 1 are as in breadth-first resolution, but generation k is produced by resolving generation k-1 only against generation 0. This has two benefits. First, the number of resolvents per generation is smaller; and second, one of the components of each resolution is fixed. Optically, this would permit generation 0 to be encoded in a quasi-static mask such as a liquid crystal mask, rather than requiring an active optical source. Unfortunately, linear-input-form resolution is not guaranteed to find the answer, even if it exists.

In *set-of-support* resolution, generation 0 contains the original facts and rules, *exclusive* of the negated hypotheses. Generation k is produced by resolving generation k-1 against the union of generations 0, 1 . . . k-1. This strategy *will* prove the hypothesis if it is true, but will produce fewer resolvents per generation. More generations may be required than in breadth-first resolution, but the size of the generations is more manageable.

A last possible form of resolution is *ancestry-filtered* resolution. In this form, generation 0 is the original set of facts and rules, and generation 1 is chosen in any of the ways previously described. However, each resolvent in generation 1 is associated with a table of 2 clauses which are its ancestors. More generally, in generation k, each resolvent is associated with a table of 2^k clauses which are all its ancestors. Generation k is formed by resolving each clause in generation k-1 with either the base set (generation 0) as in linear input form, or with its own ancestors from its associated table. This strategy is guaranteed to yield a solution if one exists, but requires more complex control than other strategies. Also, the length of the ancestor table for each resolvent doubles each generation.

OPTICAL DATA REPRESENTATIONS

In the propositional calculus, clauses may be conveniently represented by bit arrays, since each possible constant literal may occur at most once in each clause. We may denote each possible constant by a bit pair, with 00 = not occurring, 01 = asserted, and 10 = negated. The fourth possibility, 11, can be used during resolution to detect contradiction. Figure 1 shows the representation of a sample pair of clauses in the propositional calculus. The length of the bit vector representing a clause is twice the number of total constants required in the problem. In an optical implementation, the dimension of the bit vector may be either time or space. Presenting all components of the vector simultaneously has the advantage that less optical memory is required during the resolution process, and that the time domain is preserved for holding the growing number of resolvents produced in each new generation. Presenting the

components of the vector serially in time requires less transverse connection in the optical hardware.

The optical implementation could take several forms. To illustrate one technique for performing operations in the propositional calculus, consider Fig. 2. This figure shows the dibit representation of the resolution of A + B' and B + C. The possible dibits are shown in Fig. 2(a) where the two squares associated with each possibility can be thought of as being light or dark. The two clauses in Fig. 2(b) are superimposed in such a way as to cause the light patterns to overlap as in Fig. 2(c). The contradiction is removed to produce the final answer in Fig. 2(d). The dibits of Fig. 2(a) could be represented by laser diodes or light emitting diodes in an input as shown in Fig. 3. The resolution of Fig. 2(b) could be performed as in Fig. 3 where one dibit is rotated before being spread with cylindrical lenses. The superposition achieved with the transmission mask and prisms of Fig. 3 is represented (transposed) in Fig. 1(c). Holographic elements also could be used. To remove the contradiction, either an optical-to-electrical conversion is required or, ultimately, optical logic elements could be used. One two-dimensional means of removing conflicts would be a liquid crystal array. The cells of the liquid crystal would be transmissive unless both dibits of a literal are illuminated. In that case, an AND operation would cause those cells to switch. The doubly illuminated dibit and hence the contradiction would be removed.

Resolution of the type described using Figs. 2 and 3 can be done in parallel. A bit vector representing a clause can be resolved with several clauses in parallel by use of the technique illustrated in Fig. 4. All of the optical components are not shown. The light patterns representing a series of clauses,

introduced temporally, are spread to overlap the patterns of several clauses at once. Each column in the result matrix is a separate resolvent. A spatial light modulator, such as the liquid crystal array, can then remove all contradictions in parallel. An alternate approach that allows several clauses to be resolved with respect to several others is shown in Fig. 5. In this case, the literals are not introduced in parallel, as is the case in Fig. 4, but an array of sources are modulated to introduce them serially in time. The first literals in all the clauses shown horizontally are resolved with respect to all of the first literals of the clauses shown vertically. The next clock cycle, all of the second literals are presented, etc.

Representation in the quantified propositional calculus is considerably more complicated. Here, each clause is composed of literals, each of which is an *ordered list containing* a predicate name followed by arguments. The arguments may be either constants or variables. The representation must allow the optical substitution of any constant for a variable, or one variable for another in any of the argument positions, and it must be possible to make the same substitution in all literals within the clause. The representation must allow for an indefinite number of literals in each clause because, in the quantified propositional calculus, a predicate may appear more than once in a clause, each time with different arguments. A feasible representation here is to encode each literal as a rectangular bit array, with one dimension representing the sequence of names in the ordered list, and the other dimension used for an encoding of the actual predicate or argument name. Figure 6 is an example of a literal represented in this way. Each row in this notation specifies an element of the ordered list of names which represents a literal. Row 1 encodes the predicate

name itself, and whether the predicate is asserted or negated. This row could use the same dibit notations used in the propositional calculus. Row 2 represents the first argument. Each column in Rows 2 → n represents a particular constant or variable name. Thus in each row, exactly one column will be eliminated to indicate the name of the corresponding argument of the predicate. Row k is used to encode the name of the k-1 th argument. Substitution of a constant for a variable or a variable for a variable can easily be performed by deflecting the column associated with the variable to be eliminated into the column of the constant or variable to be substituted for it. This is not as easy with more densely encoded representations.

A clause is represented by adjoining a set of literals in either space or time. The spatial representation has the problem of encountering implementation bounds if the number of literals in a clause gets large, but one can argue that it is unlikely that clauses with a large number of literals will participate in the successful solution of a problem in which the objective is to reduce the number of literals to zero! A time serial representation makes it easier to handle large numbers of literals, but also makes the performance of parallel operations more difficult.

In view of the complexity introduced by moving from the propositional calculus to the quantified propositional calculus, it is not surprising that moving to the first order predicate calculus introduces another quantum leap in complexity. By allowing the arguments of predicates to be functions, which themselves have arguments (which may also be functions), the matrix notation proposed above is completely confounded. Conventional expert systems use a graph-structured notation for such complex clauses, and there is no obvious

way to translate this notation into a parallel form which can be optically manipulated. This problem must be left until the state of the art in integrated optics and optical control has taken another giant step.

OPERATIONS

The process of resolution can be regarded as a set of nested loops, in the computer science sense. The outermost loop is the process of producing successive generations of resolvents using whatever resolution strategy is selected. The next innermost loop is the process of producing the resolvents themselves through the interaction of earlier resolvents. This process may be serial or parallel, or both, depending on how the implementation represents resolvents, and whether growth in the data set is handled in time or in space. In the quantified propositional calculus, there is an additional innermost loop running through the possible combinations of literals within two clauses which might be capable of unification with proper substitutions. Within these loops are contained the actual operations of resolvent formation, substitution and union of clauses. We indicate below how these operations may be performed optically. At the present state of technology, some or all of the interactive control required for a complete system would have to be supplied with electro-optic interfaces, but the operations themselves are amenable to optical implementation.

The fundamental operation in the propositional calculus is resolvent formation. Using the dibit representation introduced earlier, a trial resolvent of two clauses can be formed merely by optical superposition, as shown in Figure 3. A matrix of trial resolvents can be formed by spreading a clause in one

dimension with a cylindrical lens and superposing it upon a matrix of clauses. Each row in the result matrix is a potential resolvent. The validity of a particular trial resolvent can be ascertained by counting the number of dibits having a value of 11 in the result. These dibits correspond to a literal which was asserted in one clause and negated in the other. The fundamental role of resolvent formation states that a valid resolvent exists if and only if precisely one of these occurs in a trial resolvent.

It is not necessary to actually count the number of contradictions in a trial resolvent. We may introduce a pair of boolean control variables C and X, such that C is true if at least one contradiction has been detected in a set of literals, and X is true if more than one contradiction has been detected. These variables can be determined in $\log_2(k)$ logic levels for k possible literals using the algebraic formulation of Eqs. (1) through (4). In this formulation, j is the logic level, and the outcome for a valid resolvent is $C[final,0] = 1$ and $X[final, 0] = 0$.

$$C[1, i] = \{1 \text{ if } dibit = 11 \text{ } else \text{ } 0\} \tag{1}$$

$$X[1, i] = 0 \tag{2}$$

$$C[j, i] = C[n-1, 2i] \quad or \quad C[n-1, 2i + 1] \tag{3}$$

$$\begin{aligned} X[j, i] = (&C[n-1, 2i] \quad \textbf{and} \quad C[n-1, 2i+1]) \\ or \quad &X[n-1, 2i] \quad or \quad X[n-1, 2i+1] \end{aligned} \tag{4}$$

Optically, this can be implemented with logic operating in one dimension upon adjacent elements representing a trial resolvent. To manage the data set effectively, the outcome should be used to drive an optical compaction system, perhaps using light values and acousto optic deflectors or a holographic array, to 'squeeze out' invalid resolvents and maintain the physical adjacency of valid

resolvents. This can be accomplished using $\log_2(n)$ levels of deflectors, where n is the number of trial resolvents generated in parallel. This is shown in Fig. 7.

In this example, the resolvents in positions 1, 4, and 6 are invalid. The system of deflection removes these and presents resolvents 2, 3, 5, 7, and 8 in a physically adjacent pattern at the output.

Once we move to the quantified propositional calculus, the operations become much more difficult. To resolve two clauses, not only the original clauses must be examined, but the permissible unifying substitutions must also be performed. These substitutions can be considered by the following implementation.

For two clauses, each containing N literals (predicate functions), there are N^2 possible ways to resolve them, each using a different pair of literals. This suggests that each clause be replicated along a row or column, and that each of the elements in the matrix produced by superimposing these rows and columns is a potential literal to be eliminated by resolution. This replication and superposition of clauses could be done with a holographic array simulating lenses and prisms. The internal structure of each element is as shown in Fig. 6, while the overall matrix of elements is arranged as shown in Fig. 8. Each element in Fig. 8 represents a comparison of a literal in one clause and a literal in another. For example, the upper left hand element of Fig. 8 represents the unifier of $P(x, A)$ and $Q(z, B)$. Obviously this unification fails. However, the center element, representing the unification of $P'(x, y)$ and $P(x, z)$, succeeds with the substitution of y for z (or z for y). The comparison is *valid* if the predicate name is the same for both literals, but asserted in one and negated in the other, and the arguments are compatible. Arguments are compatible if they are

either the same, or can be made the same be substituting constants for variables or variables for variables. At each matrix element in Fig. 8, different substitutions might be required to resolve the two clauses, but if each literal is a predicate of K arguments, at most K substitutions need be made. This can be implemented by passing the elements, in parallel, through K optical substitution planes, each of which considers a different argument to the predicate function. The following operations must be performed in each plane, at each element:

If both arguments are constants, propagate the union of their values. (5)

Propagate the constant for all occurences of the variable if *one argument is a variable.* (6)

If both arguments are variables, propagate one of them. (7)

The optical output from each plane is the input to the next plane. In the final output, a successful resolution will be indicated by an element with variable arguments (outputs in the variables portion of the pattern of Fig. 6) or a single non-conflicting constant in each argument position, and identical but complementary predicate names, similar to the situation in the propositional calculus.

The algorithm described above detects the possibility of valid resolvents, but does not actually generate them. To produce the actual resolvents, the resulting matrix must be scanned for valid elements, and for each one found, the corresponding resolvent generated. If we optically shift the clauses as the result matrix is scanned, we can generate shifted versions of the clauses such that the literal to be resolved out is the first literal in one clause and the last literal in the other clause. This is shown in Fig. 9. The valid resolvent can

then be formed by juxtaposing the two clauses after performing the same substitutions on the entire clause pair that were performed on the literals using operations 5 through 7 above. Thus, the clauses shown in Fig. 9(b) were generated by circularly right shifting the original clauses of Fig. 9(a) one position right. The clauses in Fig. 9(c) were formed from the clauses in Fig. 9(b) by uniformly substituting A for x and w for y. The resulting clauses are then resolved by eliminating the complementary literals to form the resolvent of Fig. 9(d). Since several elements of Fig. 8 may allow resolution, this process may have to be performed several times to generate all the valid resolvents of two clauses in the quantified propositional calculus.

The shifting is easily implemented using fixed optical elements. Once the literals to be removed in the removal of contradictions have been shifted to one end, an imaging system can combine the two clauses. Appropriately placed stops will remove the contradictions.

As in the propositional calculus, resolution in the quantified propositional calculus requires the consideration of all possibilities allowed by the resolution strategy selected. In the quantified propositional calculus, the consideration of a single possibility is complex enough that the outer loops of the process will probably be implemented electronically. Furthermore, the nature of the processes is such that some inputs to the system change much more rapidly than others. This suggests the possibility of using Fredkin gates [8, 9] for much of the optical logic, because one of their characteristics is an asymmetry in time response between inputs. By using the slow input for less frequently modified data, the performance of the system could be preserved while taking advantage of the simplicity of the technology.

SUMMARY

We have described the process of mathematical resolution for the propositional calculus, quantified propositional calculus, and first order predicate calculus. For the first two of these, optical data representations have been proposed. A feasible optical implementation of resolution has been presented for the propositional calculus, and its extension into the quantified propositional calculus has been discussed. The implementation in the quantified propositional calculus is sufficiently complex that significant electronic and electro-optical assistance will be required. Nonetheless, the implementation is significantly more parallel than present methods of performing resolution, and offers promise for higher performance in expert systems.

ACKNOWLEDGEMENT

This work was supported in part by the Air Force Office of Scientific Research.

REFERENCES

1. "Special Issue on Optical Computing", *Proc. IEEE* **72, 758-974 (1984).**

2. "Special Issue on Optical Computing," *Opt. Engineer* **24**, 1-91 (1985).

3. "Special Issue on Digital Optical Computing", *Opt. Engineer* **25**, 1-116 (1986).

4. "Special Issue on Optical Computing, Part I: Digital Optics", *Appl. Optics* **25**, 1511-1621 (1986).

5. Winston, P.H. and Horn, B.K.P., *LISP* (Addison Wesley, Reading, MA, 1981).

6. Nilsson, N., *Principles of Artificial Intelligence* (Tioga, Palo Alto, CA, 1980).

7. Winston, P.H., *Artificial Intelligence* (Addison Wesley, Reading, MA, 1984).

8. E. Fredkin and T. Toffoli, "Conservative Logic", *Int. J. Theoret. Phys.* **21**, 219 (1982).

9. J. Shamir, H.J. Cauldfield, W. Micelli, and R.J. Seymour, "Optical Computing and the Fredkin Gates", *Applied Optics* **25**, 1604 (1986).

FIGURE CAPTIONS

Figure 1. Sample clauses in the propositional calculus. The possible literals are A, B, C, D.

Figure 2. Dibit representation and resolution of two clauses. (a) dibit representation, (b) the two clauses A + B' and B + C, (c) trial resolvent, (d) completed resolution.

Figure 3. An optical method for resolution using dibits.

Figure 4. Spatially parallel resolution of one clause and a set of clauses.

Figure 5. Time serial resolution of literals associated with two sets of clauses.

Figure 6. Structure of the literal $P'(A, y, C)$ in the quantified propositional calculus.

Figure 7. Resolvent compaction illustrated.

Figure 8. Replication of clauses.

Figure 9. Sequence of operations in clause unification. (a) original clauses, (b) clauses after shifting, (c) clauses after substitution, (d) unified and resolved clauses.

$$A + B' + D$$

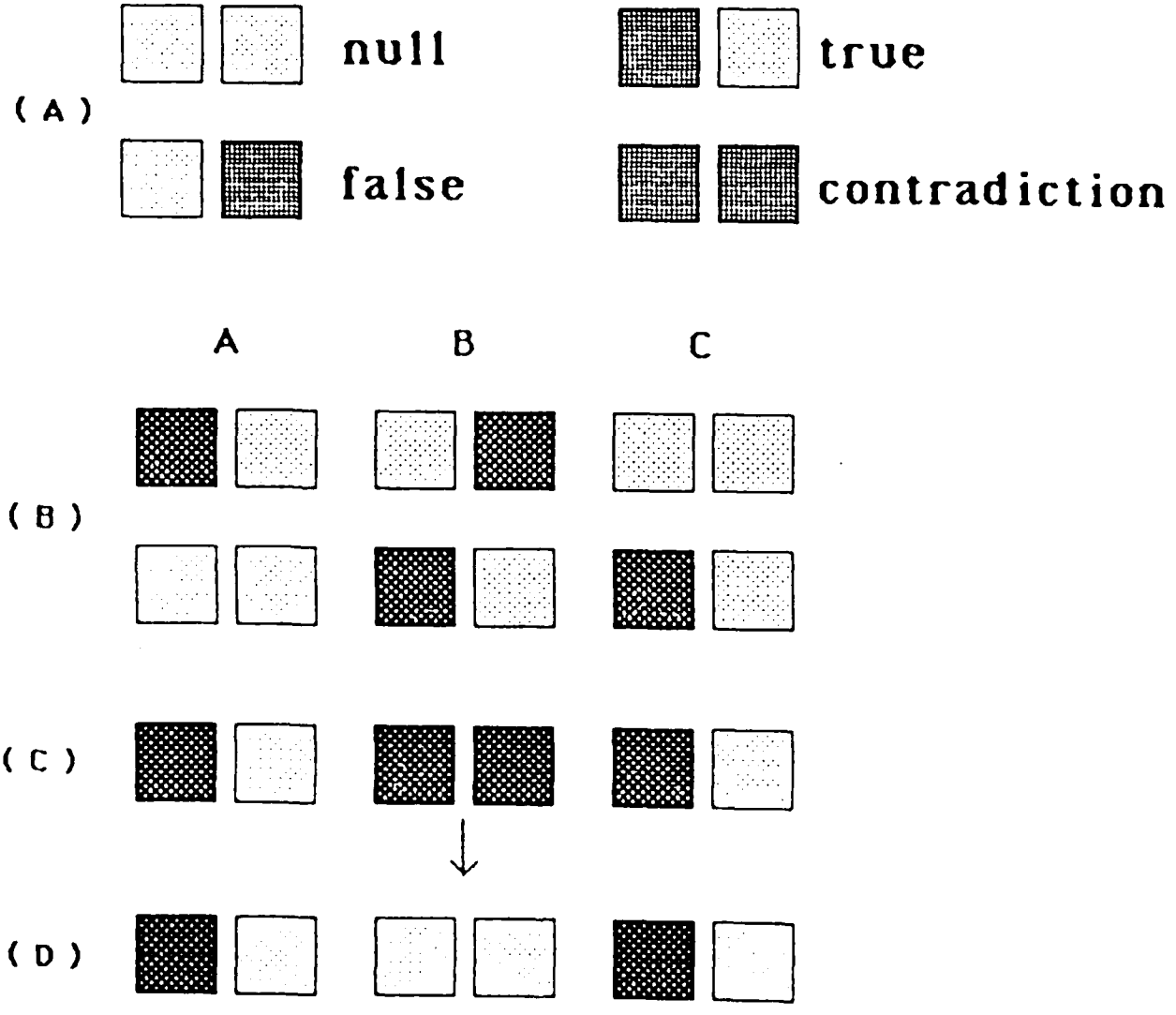
10010010

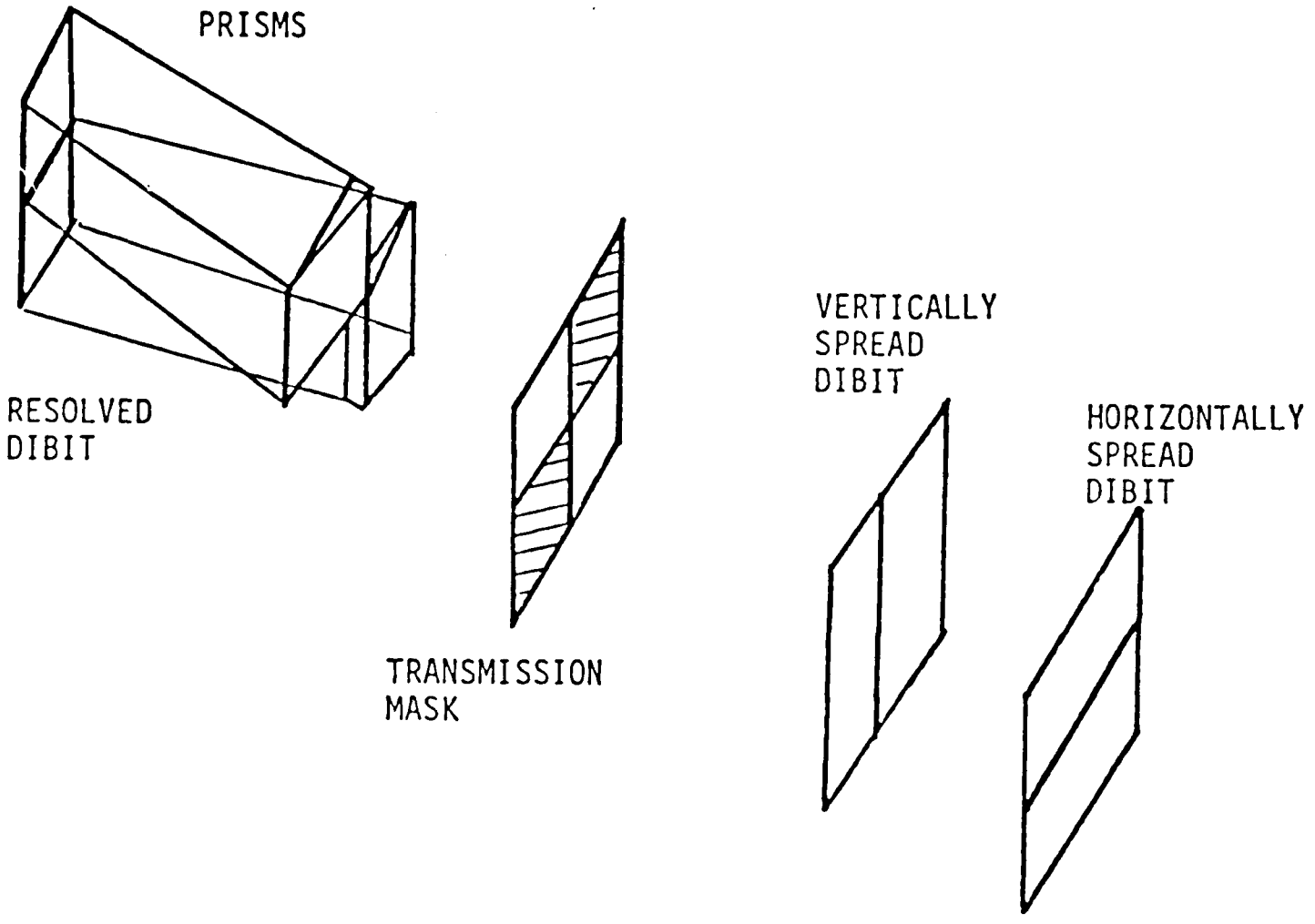
$$B + D$$

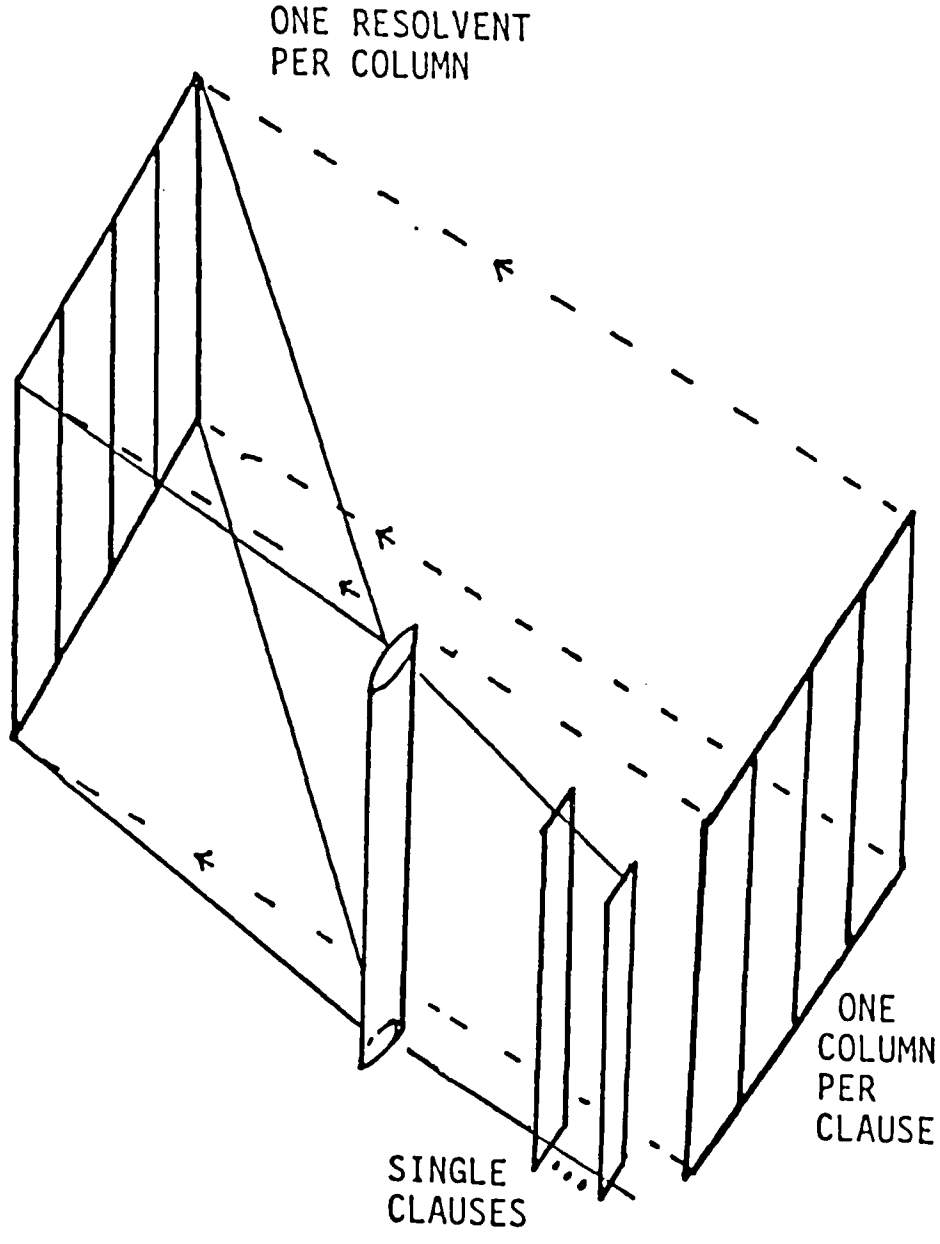
00100010

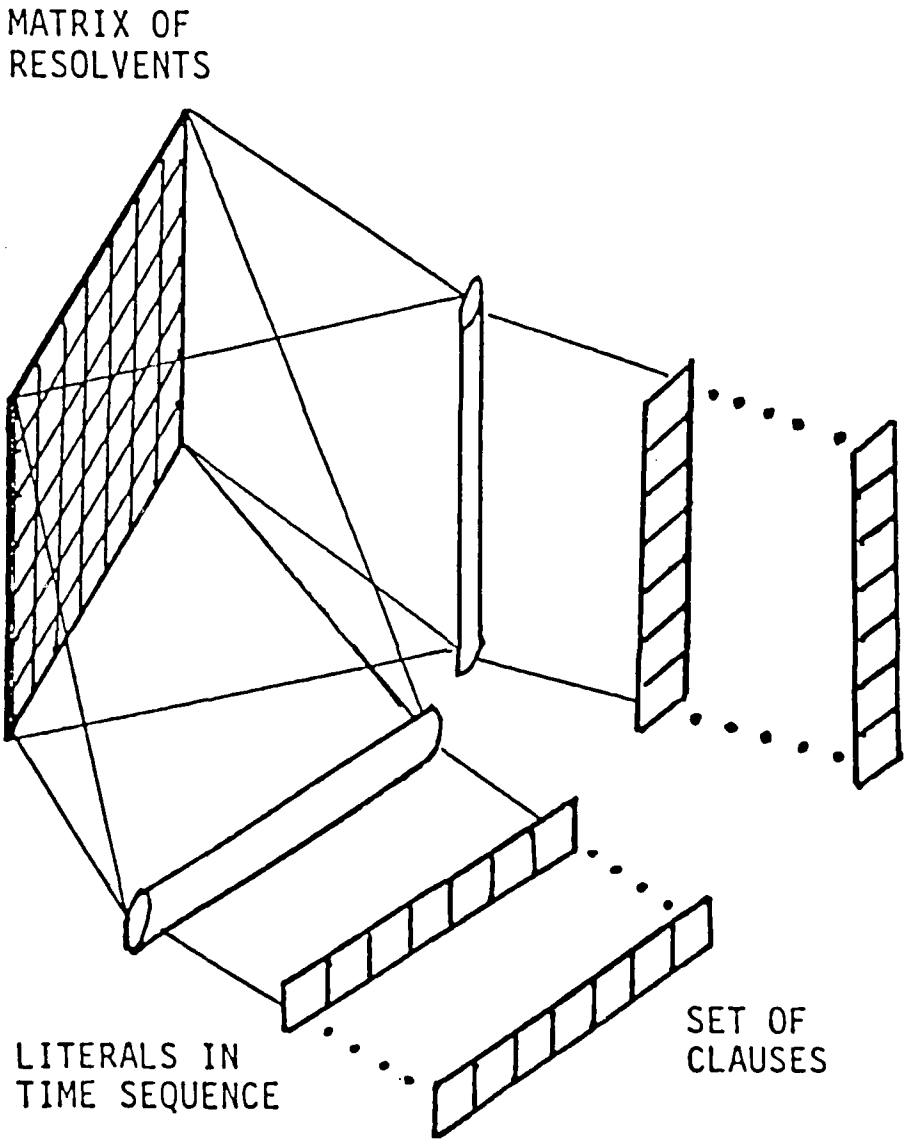
$$A + D$$

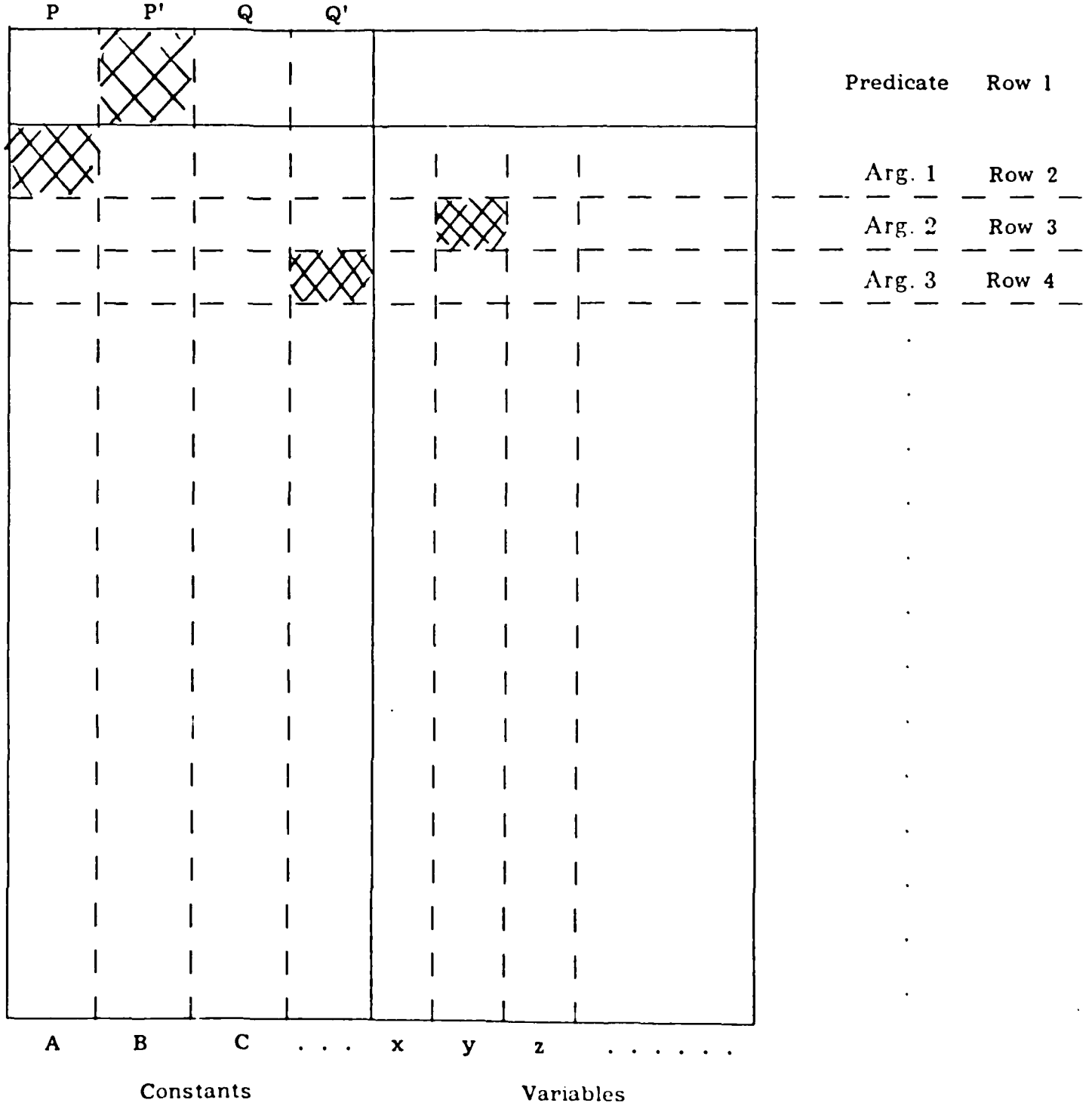
10000010

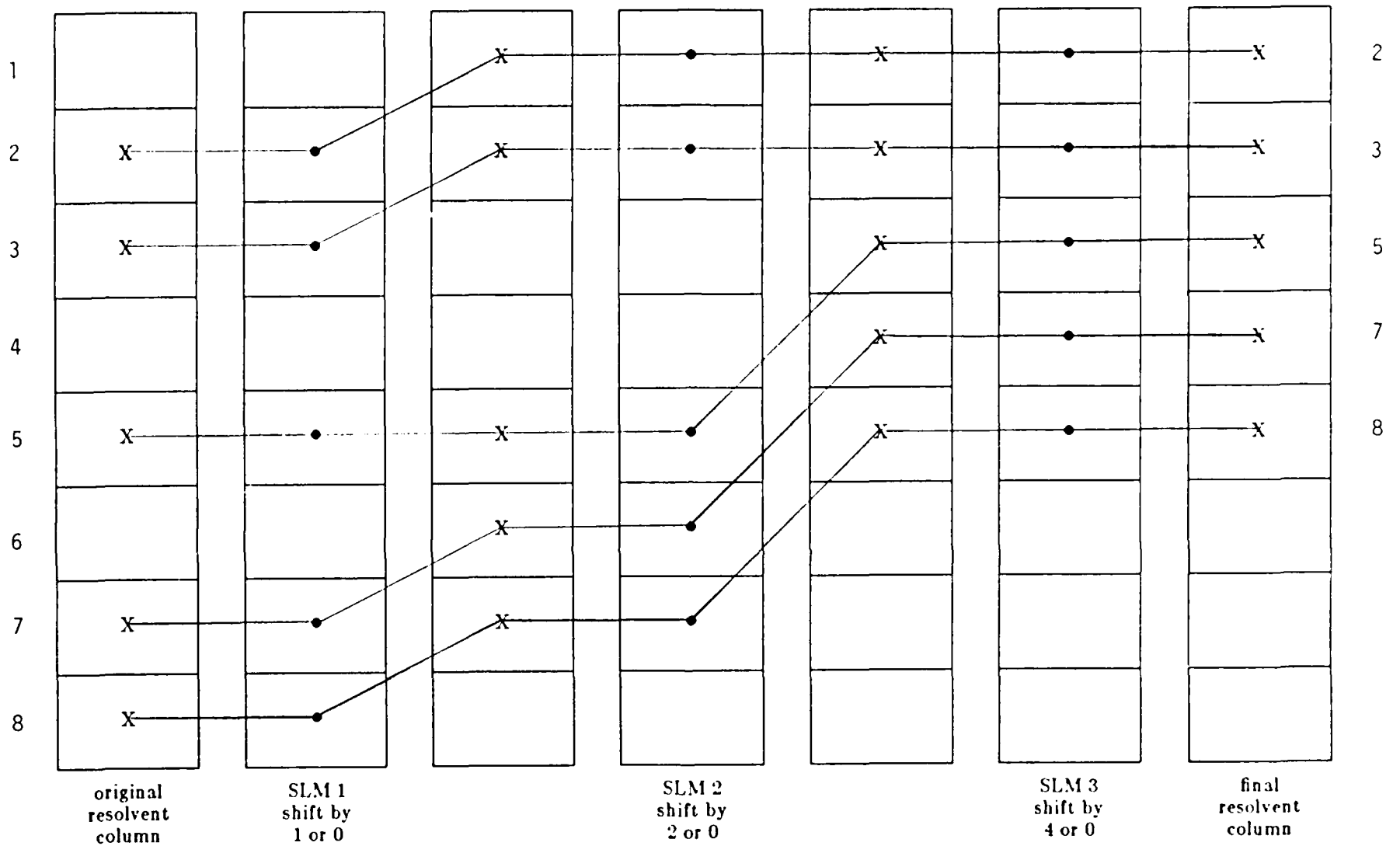

(A)
null
true
false
contradiction
A
B
C
(B)
(C)
(D)

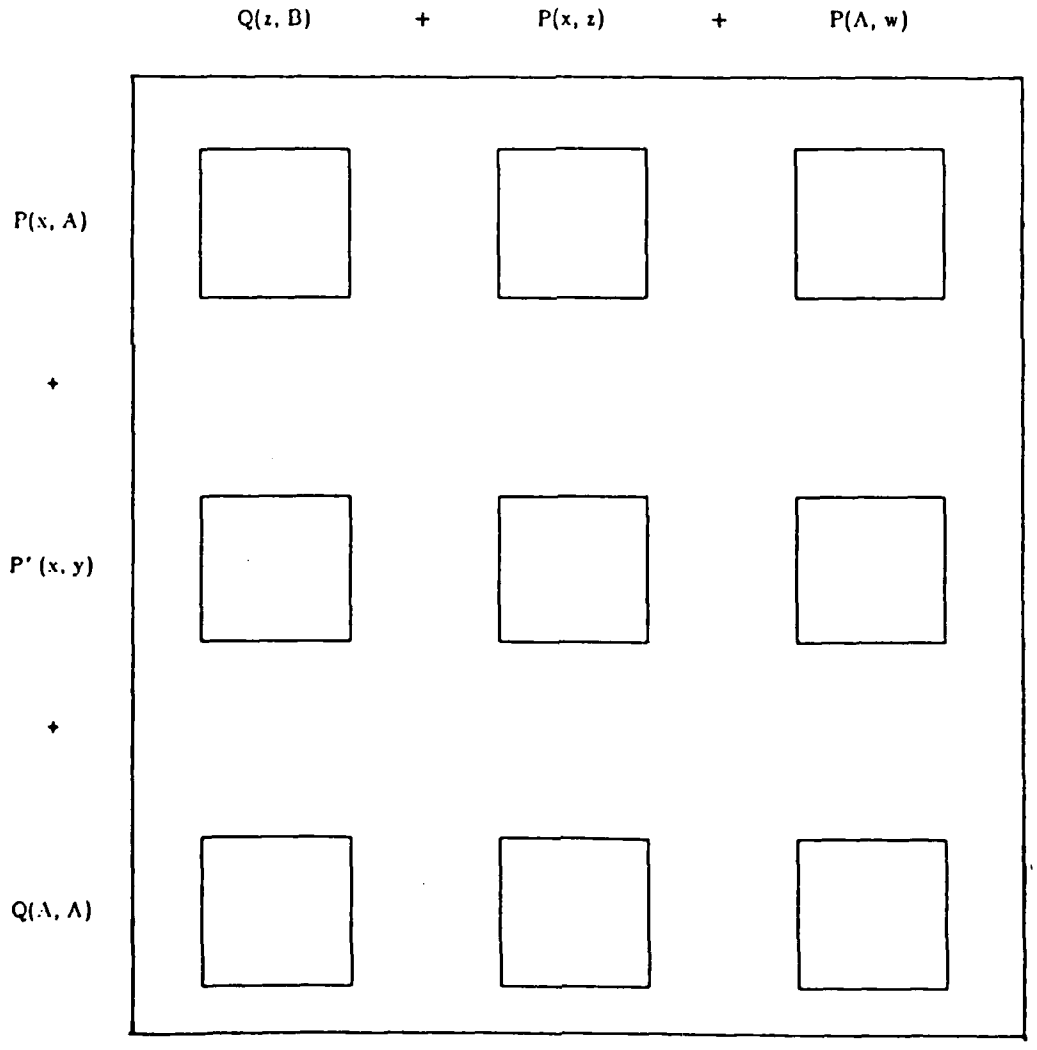

PRISMS
RESOLVED
DIBIT
TRANSMISSION
MASK
VERTICALLY
SPREAD
DIBIT
HORIZONTALLY
SPREAD
DIBIT


ONE RESOLVENT
PER COLUMN
ONE
COLUMN
PER
CLAUSE
SINGLE
CLAUSES


MATRIX OF
RESOLVENTS
LITERALS IN
TIME SEQUENCE
SET OF
CLAUSES


P
P'
Q
Q'
Predicate
Row 1
Arg. 1
Row 2
Arg. 2
Row 3
Arg. 3
Row 4
A
B
C
. . .
x
y
z
. . . . . .
Constants
Variables


1
2
3
4
5
6
7
8
2
3
5
7
8
original resolvent column
SLM 1 shift by 1 or 0
SLM 2 shift by 2 or 0
SLM 3 shift by 4 or 0
final resolvent column
X = good resolvent


Q(z, B)
+
P(x, z)
+
P(A, w)
P(x, A)
+
P' (x, y)
+
Q(A, A)

(a) $P(x, A) + P'(x, y) + Q(A, A)$

$Q(z, B) + P(w, z) + P(A, w)$

(b) $Q(A, A) + P(x, A) + P'(x, y)$

$P(A, w) + Q(z, B) + P(w, z)$

(c) $Q(A, A) + P(A, A) + P'(A, w)$

$P(A, w) + Q(z, B) + P(w, z)$

(d) $Q(A, A) + P(A, A) + Q(z, B) + P(w, z)$

MATRIX-VECTOR MULTIPLICATION USING POLARIZATION ROTATORS

L. Scharf, W. T. Cathey, K. M. Johnson

Center for Optoelectronic Computing Systems

Electrical and Computer Engineering Department

University of Colorado, Boulder

Boulder, CO 80309-0425

Abstract

A new approach to optical matrix-vector multiplication is described which matches signal processing algorithms and architectures to optical primitives which directly perform rotation operations.

MATRIX-VECTOR MULTIPLICATION USING POLARIZATION ROTATORS

L. Scharf, W. T. Cathey, K. M. Johnson

Center for Optoelectronic Computing Systems
Electrical and Computer Engineering Department
University of Colorado, Boulder
Boulder, CO 80309-0425

Introduction

The potential application of optical systems to perform high speed, low cost signal processing with large parallelism has attracted the attention of researchers for many years. General optical processors have been developed that compute matrix-vector multiplications and other linear algebraic operations using incoherent light. One example is the Optical Matrix-Vector Multiplier (OMVM), which calculates the discrete operation of a matrix-vector product, rather than the continuous correlation and convolution more commonly associated with optical processing [1]. The OMVM can be used to compute discrete Fourier transforms (DFT's), and for performing linear algebraic operations, including matrix-matrix multiplications. It has been suggested as a method for implementing associative memory [3-5] and optical crossbars [4]. The first OMVM had several disadvantages, including low accuracy, low speed, and a nonprogrammable matrix mask. Recent implementations use real-time spatial light modulators (SLM) [5-7] and acousto-optic cells [8]. The two-dimensional spatial light modulators used in many of these optical processors operate at millisecond speeds, are expensive and have low resolution [5, 7]. One-dimensional modulators such as acousto-optic cells are faster, but the major drawback of computing matrix-matrix operating using one-dimensional devices is that to calculate two-dimensional matrix-matrix operations, data from the rows and columns of matrices must be loaded serially. The cycle time through the processors increases with the order of the matrix, and the natural parallelism of optics is lost.

Objective

The goal of our research is to achieve 100 x 100 matrix-matrix multiplications in a microsecond, with 10 bit or greater accuracy. To achieve this goal, a new approach is needed. We describe a two-dimensional optical systolic processor with new algorithms, architectures, and devices which we believe will result in the evolution of an optical processor capable of meeting this goal. In this paper we outline our design principles for high-speed, high precision optical implementations of linear algebraic computations.

One can view the matrix-matrix multiplications problem with the frame work of an I/O problem and a realization problem.

(i) I/O problem : multiply matrices **A** and **B**.

For this I/O problem there are an infinite number of *realizations* or *algorithms* that one can use to perform the multiplications.

We can use this freedom to optimize criteria associated with the computation. For example, in some digital processing problems we choose an algorithm to minimize the number of computations. In this particular application we wish to design algorithms which use low accuracy primitives to obtain a high accuracy result. We also wish to pipeline computations, develop highly regular and locally connected geometrices, and to use simple optical primitives as the basis of the algorithms.

(ii) Realization problem.

The realization problem consists of finding architectures that consist of simple optical primitives, connected in modular geometries, to produce high-accuracy results by pipelining the computations through low accuracy cells. This goal involves:

(a) low accuracy primitives for high accuracy results
(b) modular geometrices
(c) pipeline computations
(d) simple, optical primitives

Algorithms and Architectures

The algorithms being used for this processor break-up matrices into repetitive operations on a smaller set of orthogonal rotation matrices. The algorithms are low loss and the architectures used to implement the algorithms are cellular, as shown in Fig. 1, and based on optical operations [9].

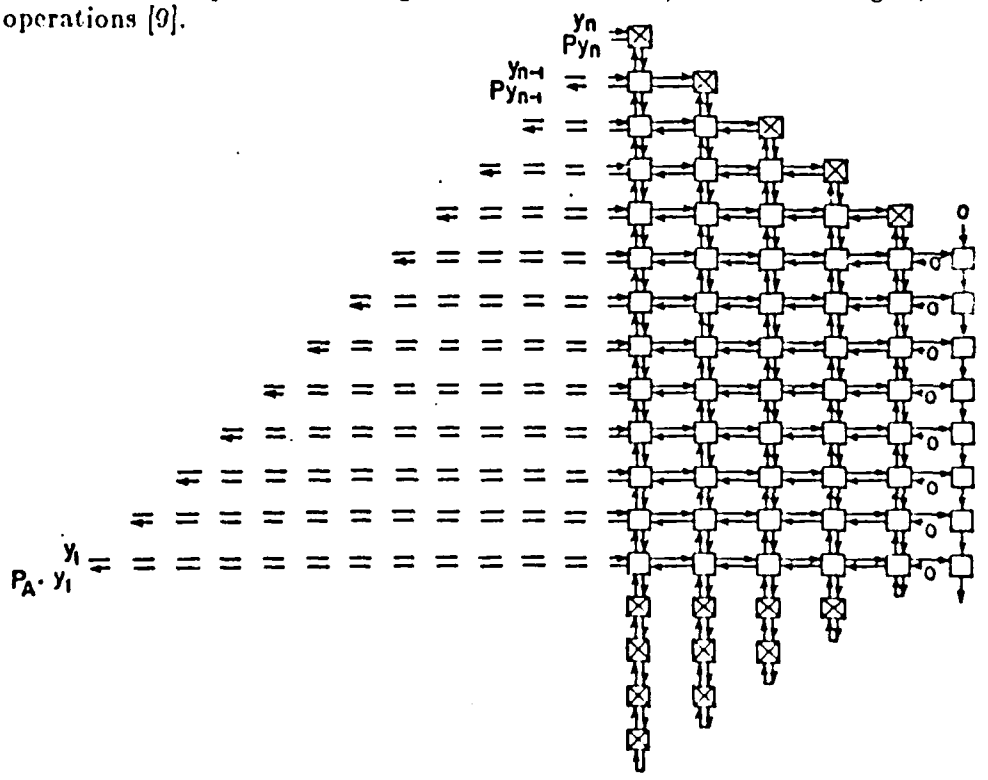


Figure 1. Cellular Implementation of a Vector Pipelined Projection Operator.

We can use this freedom to optimize criteria associated with the computation. For example, in some digital processing problems we choose an algorithm to minimize the number of computations. In this particular application we wish to design algorithms which use low accuracy primitives to obtain a high accuracy result. We also wish to pipeline computations, develop highly regular and locally connected geometrices, and to use simple optical primitives as the basis of the algorithms.

(ii) Realization problem.

The realization problem consists of finding architectures that consist of simple optical primitives, connected in modular geometries, to produce high-accuracy results by pipelining the computations through low accuracy cells. This goal involves:

(a) low accuracy primitives for high accuracy results
(b) modular geometrices
(c) pipeline computations
(d) simple, optical primitives

Algorithms and Architectures

The algorithms being used for this processor break-up matrices into repetitive operations on a smaller set of orthogonal rotation matrices. The algorithms are low loss and the architectures used to implement the algorithms are cellular, as shown in Fig. 1, and based on optical operations [9].

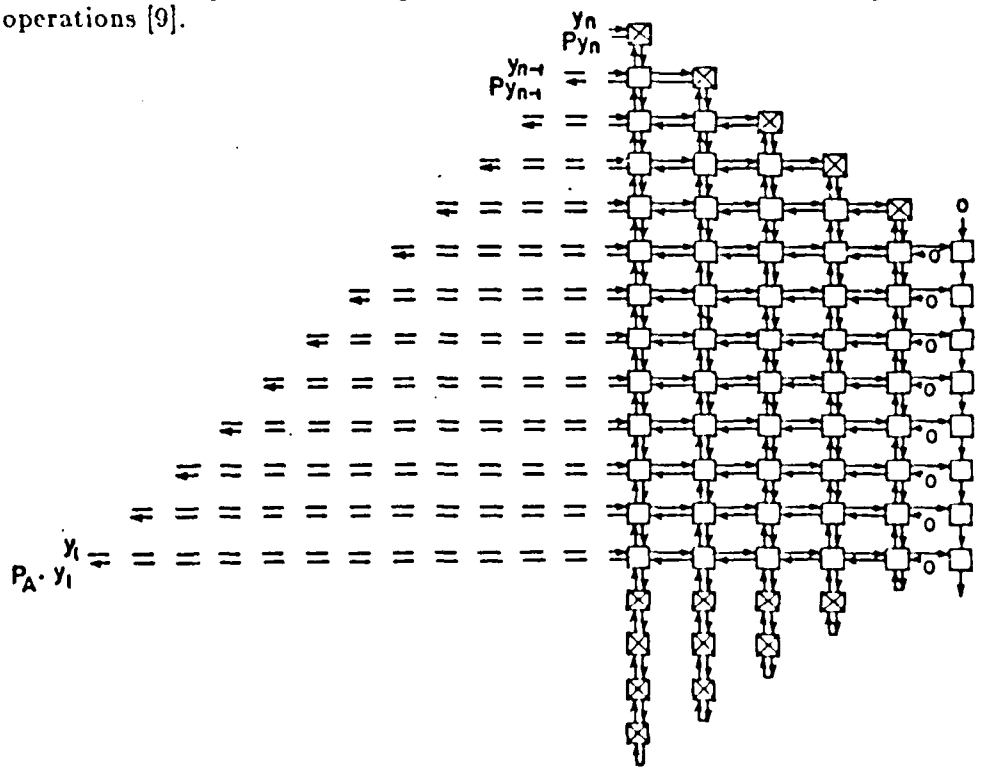


Figure 1. Cellular Implementation of a Vector Pipelined Projection Operator.

Optical Implementations

Figure 2 illustrates the rotation operation on incoming signals as a (2 x 2) matrix map. This same operation can be implemented optically using devices that rotate the polarization of the input vector. One optical implementation of the rotator-combiner is shown in Fig. 3, where the first element is a polarizing beamsplitter which separates the x and y components. The second polarizing beamsplitter acts as a combiner of the appropriate components, and a polarization rotator then imparts the desired rotation onto the resulting vector. For hard-wired applications, quartz, which gives a rotation fo 21.7° /mm, could be used. The thickness can be controlled to yield the desired rotation. Electrically controlled rotators would give programmability and an array of liquid crystals could provide discrete rotations.

$$\begin{vmatrix} V_1 \\ V_2 \end{vmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \cos\theta & \sin\theta \end{bmatrix} \begin{vmatrix} U_1 \\ U_2 \end{vmatrix}$$

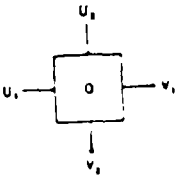


Figure 2. Signal Rotation Operation.

Figure 4 shows that, with the development of a rotator-combiner cell, the general problem of implementing matrix-vector and matrix-matrix multipliers in numerically stable machines can be implemented in a regular cellular array of such rotator-combiner cells.

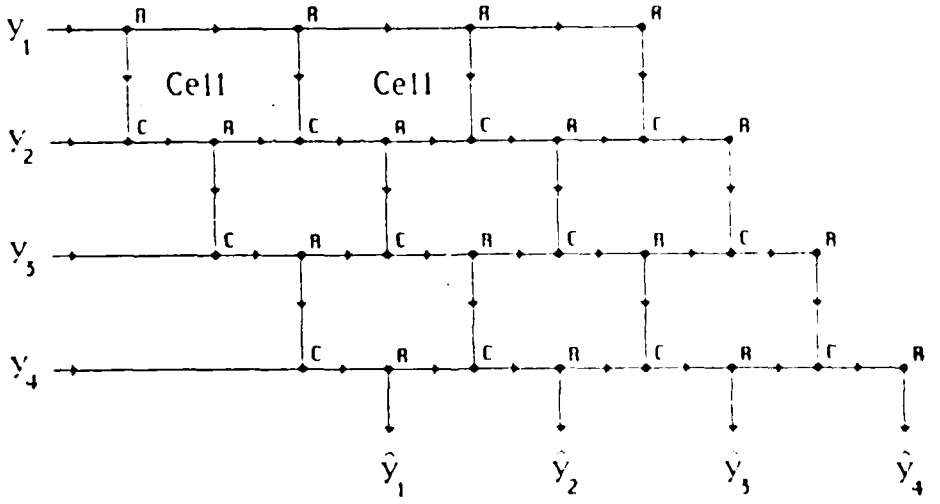


Figure 3. Cellular Architecture for Implementing a Sequence of Rotations.

We will discuss implementing the rotator-combiner cell using polarizing beamsplitters, and ferroelectric liquid crystal (FLC's) which can switch the polarization of incident light in less than a microsecond [10, 11]. These crystals, developed at the University of Colorado, Boulder, in the Physics Department have already been fabricated successfully in 32 x 32

matrix arrays [12]. By making 256 x 256 matrix arrays, a trade-off between array size and accuracy can be achieved. In addition, since these FLC's are capable of submicrosecond switching speeds, a trade-off between speed and accuracy can now be made for the first time.

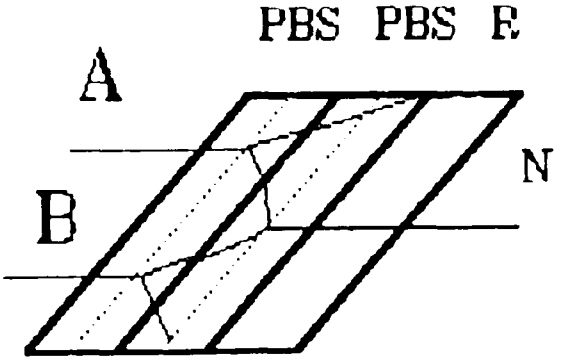


Figure 4. Integrated Rotator-Combiner Using Polarizing Beamsplitters.

References

(1) Goodman, J. W., Dias, A. R., Johnson, K. M., and Peri, D., "Parallel Incoherent Optical Matrix-Vector Multipliers," Proc. Workshop on Optical Signal Processing, Texas Tech. University, Lubbock, Texas, 116-128 (1980).

(2) Goodman, J. W., Johnson K. M., "The Incoherent Matrix-Vector Multiplier," Final Report (L7-1-1, April (1981).

(3) Psaltis, D. and Farhat, N., "Optical information Processing Based on an Associative Memory Model of Neural Nets with Thresholding and Feedback," *Opt. Lett.* **10**, 98 (1985).

(4) Bell, T.L A., "Optical Computing: A Field in Flux," *IEEE Spectrum* **23**, 34 (1986).

(5) Athale, R., Lee, J. N., "Optical Processing Using Outer-Product Concepts," *Proc. IEEE* **72**, 931 (19084).

(6) Minemoto, T., Okamoto, K., Miyamoto, K., "Optical Parallel Logic Gate Using Spatial Light Modulators With the Pocket Effect," *Appl. Opt.* **24**, 2055 (1985).

(7) Grinberg, J., Soffer, B. H., Owechko, Y., and Marom, E., "Programmable Real-Time Incoherent Matrix Multiplier for Optical Processing," in Tech. Digest on Optical Computing, Paper TuD-1, OSA (1985).

(8) Verber, C. M., "Integrated-Optical Approaches to Numerical Optical Processing," *Proc. IEEE* **72**, 912 (1984).

(9) Rialan, C. P. and Scharf, L. L., "Cellular Architectures for Implementing Projection Operators," to be published in *IEEE Trans. ASSP.*

(10) Clark, N. A., Lagerwall, S. T., "Submicrosecond Bistable Electro-Optic Switching in Liquid Crystals," *Appl. Phys. Letters* **36**, 899 (1980).

(11) Clark, N. A., Lagerwall, S. T., "Physics of Ferroelectric Fluids: The Discovery of a High-Speed Electro-Optic Switching Process in Liquid Crystals," *Rec. Dev. in Cond. matter Physics* **4**, 309 (1981).

(12) Pagano-Stauffer, L. A., Johnson, K. M., Masterson, H., Handschy, M., Clark, N., "Optical Parallel Processing Using Ferroelectric Liquid Crystal Arrays," *SPIE Conference*, August (1986).

POLARIZATION-BASED OPTICAL PARALLEL LOGIC GATES USING FERROELECTRIC LIQUID CRYSTAL SPATIAL LIGHT MODULATORS

K. M. Johnson, M. Handschy, W. T. Cathey, N. Clark, D. Walba

Center for Optoelectronic Computing Systems
University of Colorado
Boulder, Colorado 80309-0425

Introduction

Optical computing systems offer an increased information processing rate by facilitating parallel computing architectures. Previous experience with electronic computers indicates that desired accuracy can be achieved only with digital computation. Since the simplest digital arithmetic is binary, most recent work on optical computing is focused on the construction of binary optical logic gates. Many practical implementations of such logic gates have been suggested, a recent review is given by Sawchuck and Strand [1]. Most previous schemes operate on light intensity, much in the way that electronic systems operate on voltage or current. Another natural optical scheme represents the two binary states with two orthogonal polarizations of light. The optical element necessary to implement this scheme is a device with two states, one of which passes light of a chosen polarization unchanged, and the other of which converts light of the chosen polarization to its orthogonal complement. Tsvetkov et al. [1] have described a practical implementation of this logic using the now common twisted nematic (TN) liquid crystal device, which has two voltage-selected states, one of which rotates the polarization direction of appropriately oriented linearly polarized light by 90° and the other of which has no rotary power. Another implementation would use any of the variable retardation effects such as the Pockels effect. One state of the device would be chosen to have zero retardation, and the other to have half-wave retardation. In addition to either passing unchanged or imparting 90° rotation to linearly polarized light, this scheme could also work by either passing unchanged or reversing the handedness of circularly polarized light. An advantage pointed out by Lohmann [3] that any implementation of polarization-based logic has over logics based on intensity is that no light is lost in the logical operation of inversion. In intensity-based logics, it is difficult to invert an already dark input, since light has to be "recreated"; polarization-based elements, as described above, can convert the light representing either logical state to the other, making easy the realization of any desired Boolean function.

We describe below a third implementation, in which the optical element is a ferroelectric liquid crystal device that functions as a half-wave plate whose axis can be electrically toggled between two orientations that make a 45° angle to each other. These elements have extremely useful operating characteristics for optical parallel processing, including fast response time (submicrosecond), low-power, low-voltage switching (tens of Volts), and bistability [4]. FLC elements have already been used in an intensity-based logic scheme, where their high contrast (up to 1500) has been exploited to advantage [5]. The polarization-based gate can perform all 16 Boolean logic functions possible with two binary inputs, without the need to manually remove or change any of the optical elements. In particular, we show especially simple implementations of the XOR and XNOR logical operations.

FLC Electrooptics

Ferroelectric liquid crystals possess properties especially attractive for optical logic applications when used in the so-called surface-stabilized geometry, which has been described extensively elsewhere [6, 7, 8]. Briefly, the FLC is disposed between two closely spaced glass plates, coated on their inner surfaces with a transparent electrical conductor. The FLC material itself is optically uniaxial (we ignore a weak biaxiality), with the uniaxis coupled to the ferroelectric polarization $\vec{P}$ so that when $\vec{P}$ is perpendicular to the glass plates, the uniaxis is parallel to them. Two such orientations of $\vec{P}$ are easily selected by voltages applied across the transparent electrodes; $\vec{P}$ prefers to be parallel to the resulting electric field $\vec{E}$. The optic axis states selected by applied voltages of opposite sign, while both parallel to the plates, differ in orientation by an angle $2\psi_0$, where the "tilt angle" ψ_0 is a material property determined by the thermodynamic characteristics of the FLC. Many FLC materials have ψ_0 close to 22° over large temperature ranges, allowing the optic axis to be electrically rotated through approximately 45°. If the thickness d of the FLC layer is chosen so that $\Delta n = \lambda/2$, where Δn is the FLC's birefringence and λ is the vacuum wavelength of the incident light, the FLC becomes a half-wave plate. If the polarization of normally incident light is chosen either parallel or perpendicular to one of the voltage-selected optic axis states, it will be transmitted through the FLC unaffected. The optic axis state selected by the opposite applied voltage is then 45° to either incident polarization, so that both the ordinary and extraordinary modes will be excited. For correct FLC elements thickness d at total phase shift of π will accumulate between these two modes, and the incident light's polarization will be rotated by 90°.

Beside the previously mentioned switching speed, the surface-stabilized FLC geometry offers another feature useful in optical logic systems: bistability. After either applied voltage brings the optic axis to one of its preferred orientations, that voltage may be removed without the optic axis returning to its previous state. This allows a two-dimensional array of FLC elements to be matrix addressed. For instance, if the conductors are divided on one plate into column electrodes and on the other plate into row electrodes, appropriate waveforms applied to the rows and columns would allow a selected element where a given pair of row and column electrodes overlap to be changed without disturbing any of the other elements in the array. A practical scheme for accomplishing this has been demonstrated by Wahl et al. [9], who achieved 1000:1 multiplexing. Thus, a large number of FLC elements ($1000 \times 1000 = 10^6$) can be simply fabricated on a single substrate, and driven with an economical number of electrical connections.

Ferroelectric Liquid Crystal Logic Gate

The XOR (AB' + A'B) and XNOR (AB + A'B') Boolean functions are the most difficult to implement optically using bright and dark logic. This is because light is irretrievably lost when creating not A (A') and not B (B'). Logic gates using bright and true logic, therefore, require four separate inputs; A, B, A', and B'.

With polarization logic, these functions are easily implemented using two FLC arrays, an optical controller, and an analyzer as shown in Fig. 1. In this gate light is not absorbed, and does not require regeneration.

For the XOR operation, the controller is in a non-switched state, and vertical light illuminates FLC array A. This array is a programmable matrix made up on transparent pixel elements which either rotate or do not rotate incident light (switched or not switched pixels). When vertically polarized laser light illuminates the switched pixels, the light is rotated to the horizontal polarized state. When the incident laser light illuminates non-switched pixels, no rotation occurs and vertical light is transmitted. A pattern made up of horizontal and vertical polarized light illuminates FLC array B. If either vertical or horizontal light illuminates a switched pixel in FLC B, the polarization is rotated by 90°, vertical rotates to horizontal and horizontal rotates to vertical. If light is incident on a non-switched pixel, the

transmitted light retains its polarization. The truth table in Fig. 2 summarizes the logical function. An analyzer at the output provides visual inspection of the XOR function.

To realize the XNOR, the FLC optical controller is switched which rotates the incident vertical laser light to horizontal light. The truth table for the XNOR function is also shown in Fig. 2.

Conclusions

We describe a new optical parallel logic gate implemented with spatial light modulators made of arrays of ferroelectric liquid crystals (FLC) electrooptic elements. The unique optical properties of the FLC elements make particularly simple a logic where two orthogonal polarizations of transmitted light represent the two binary states. A feature of this logic is that light need never be absorbed, allowing all 16 Boolean functions of two binary inputs to be implemented in a single gate; additionally, cascaded gates are equally feasible. FLC's also confer the advantages of submicrosecond switching speed and intrinsic two-state memory.

We will also discuss progress in synthesizing new FLC materials with faster switching speed, improved contrast ratio and temperature stability. Scattering and insertion losses, and switching energy measurements will be presented. A comparison of the FLC spatial light modulator with the deformable mirror device, the silicon PZLT, and the magneto-optic spatial light modulators will be made.

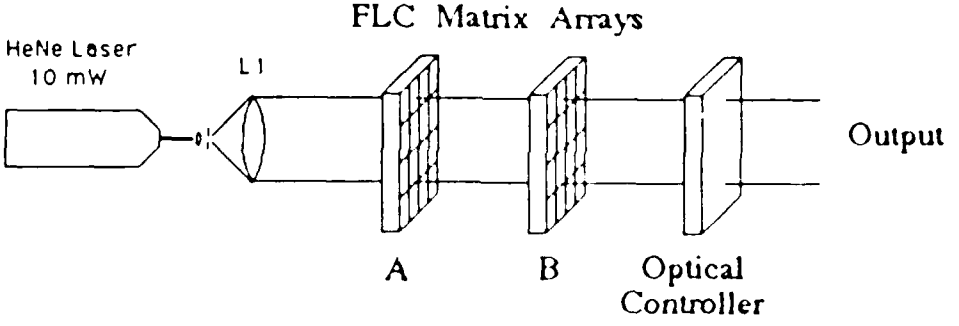


Figure 1. FLC XOR and XNOR Optical Logic Gate.

References

(1) A. A. Sawchuck and T. C. Strand, *Proc. IEEE* **72**, 758 (1984).

(2) V. A. Tsvetkov, N. A. Morozov, and M. I. Ellinson, *Sov. J. Quant. Electron.* **4**, 989 (1975).

(3) A. W. Lohmann, *OSA Annual Meeting, (1985).*

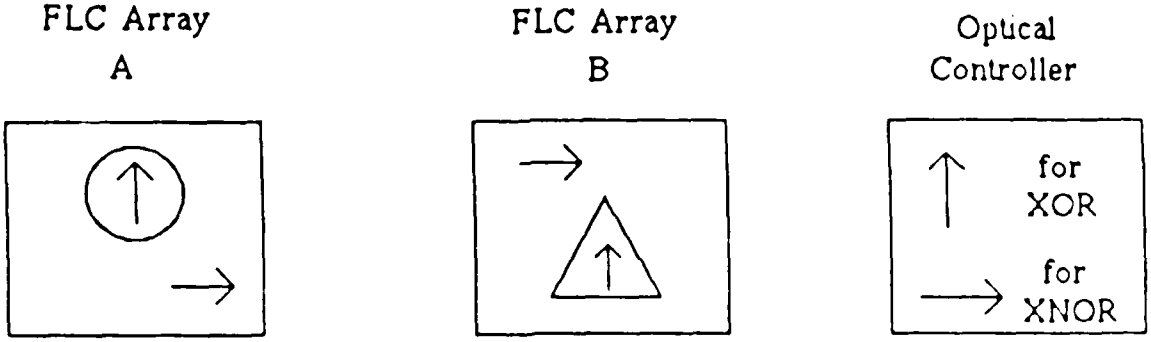
(4) N. A. Clark and s. T. Lagerwall, *Appl. Phys. Lett.* **36**, 899 (1980).

(5) L. A. Pagano-Stauffer, K. M. Johnson, N. A. Clark, and M. A. Handschy, to appear *Proc. SPIE* (1986).

(6) N. A. Clark, M. A. Handschy, and S. T. Lagerwall, *Mol. Cryst. Liq. Cryst.* **94**, 213 (1983).

(7) M. A. Handschy and N. A. Clark, *Ferroelectrics* **59**, 69 (1984).

(8) S. T. Lagerwall and J. Wahl, to appear *Mol. Cryst. Liq. Cryst.* (1986).



FLC A	FLC B	XOR output	XNOR output
↑	↑	↑	→
↑	→	→	↑
→	↑	→	↑
→	→	↑	→

Figure 2 XOR and XNOR Polarization Truth Table

754 36

Photoaddressing of High Speed Liquid Crystal Spatial Light Modulators

Garret Moddel, Kristina M. Johnson, Mark A. Handschy

Center for Optoelectronic Computing Systems
Campus Box 425
University of Colorado, Boulder
Boulder, Colorado 80309-0425

Abstract

Switching speeds of photoaddressed liquid crystal spatial light modulators are currently limited to several milliseconds. This is due in part to the choice of liquid crystal (nematic), and in part to the choice of photoaddressing schemes. In this paper we describe two methods for making photoaddressed liquid crystal spatial light modulators with microsecond response times.

Introduction

Optical computing systems offer increased information processing throughput rates by taking advantage of parallel optical architectures. The fundamental component in these architectures is a device which can modulate two-dimensional optical data. These devices are known as spatial light modulators and have many applications including input/output displays, spatial and matched filtering, incoherent - coherent light converters, optical crossbars, and optical associative memories.

Spatial light modulators (SLMs) are programmed using either electronic or optical addressing schemes. Optical addressing of spatial light modulators is advantageous because it is a direct addressing technique - a camera, frame grabber, and computer are not required to write a data pattern onto the SLM. Existing photoaddressed liquid crystal SLMs are limited to millisecond response times [1-5]. In this paper we present two methods for fabricating SLMs with microsecond response times by photoaddressing ferroelectric liquid crystals with hydrogenated amorphous silicon.

Description of the FLC Spatial Light Modulator

Ferroelectric liquid crystals possess properties especially attractive for optical processing applications when used in the so-called surface stabilized geometry [6-9]. As shown in Figure 1, a slab of essentially optically uniaxial FLC is disposed between two closely spaced glass plates, coated on their inner surfaces with a transparent electrical conductor. Voltages of opposite signal applied to the plates select between two optic axis orientations, both parallel to the plates, but differing in direction by an angle 2θ. The " tilt angle" θ, a materials property of the FLC, is close to 22.5° over large temperature ranges, allowing the optic axis to be electrically rotated through approximately 45°. If the thickness d of the FLC layer is chosen so that $\Delta n d = \frac{\lambda}{2}$, where Δn is the FLC's birefringence (typically 0.1 - 0.2) and λ is the vacuum wavelength of the incident light, the FLC becomes a half-wave plate.

If the polarization of normally incident light is chosen either parallel or perpendicular to one of the voltage-selected optic axis states, it will be transmitted through the FLC unaffected. The optic axis state selected by the opposite applied voltage is then 45° to the incident polarization, so that both the ordinary and extraordinary modes will be excited. For correct FLC element thickness d, a total phase shift of π will accumulate between these two modes, and the incident light's polarization will be rotated by 90°. Figure 2 schematically illustrates this action of the FLC electro-optic element.

The switching speed of the FLC element for a given applied electric field strength E, is largely determined by the FLC material's ferroelectric polarization P, and viscosity η, through the relation

$$\tau = \frac{\eta}{PE} \qquad (1)$$

Optical 10 - 90% times are usually about 1.8 τ [10]. The viscosity of these materials can be determined from the above relation when P is known. For the high temperature material HOBACPC, this relation gives $\eta \approx 3$ cP at the lowest temperature in its smectic C* phase [11]. For a typical room temperature material, CS-1014, one can infer $\eta \approx 50$ cP from the manufacturer's data [12]. Modest improvements over currently available polarizations should yield Ps of 1 μC/cm^2. With this polarization, applied electric fields of 100 V/μm would give switching times of about 5 ns at elevated temperature and 100 ns at room temperature. However, the speed of an FLC array is more likely to be limited by its maximum allowable power dissipation than by the switching time of its FLC material. Switching a unit area of FLC by reversing an applied voltage V dissipates an energy 2 PV through the reversal of the polarization. If this reversal is repeated as frequently as possible (i.e. once every 1.8τ), the power dissipated is 2 PV/(1.8τ) $\approx \eta d/\tau^2$, where d is the FLC thickness (E = V/d). Thus, for a given maximum allowable power dissipation W, the shortest

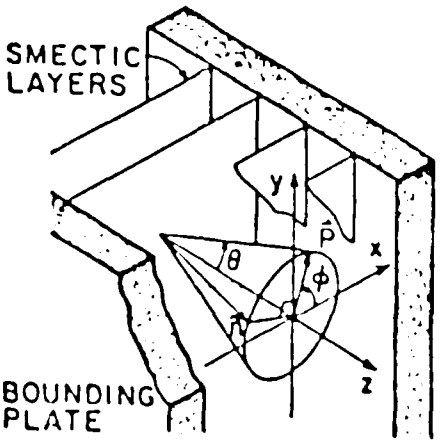


Figure 1. Ferroelectric liquid crystal structure.

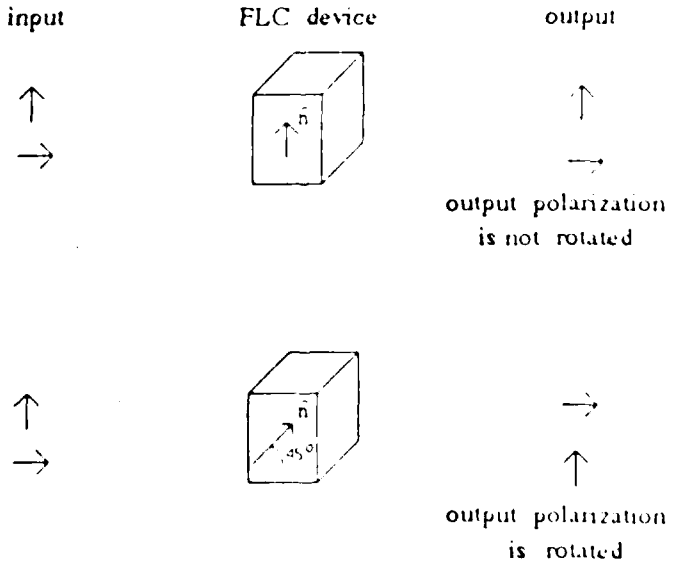


Figure 2. Smectic C phase ferroelectric liquid crystal as an addressable half-wave plate.

achievable characteristic time is given by $\tau = [\eta d/W]^{1/2}$. For operation at room temperature of an FLC device of d = 1 μm with a dissipation of W = 100 mW/cm^2, this relation gives a minimum switching time of 13μs; at an elevated temperature with W = 1 W/cm^2, the minimum switching time is reduced to 550 ns.

Because the previously mentioned switching speed, the surface-stabilized FLC geometry offers another feature useful in optical logic systems: bistability. After either applied voltage brings the optic axis to one of its preferred orientations, that voltage may be removed without the optic axis returning to its previous state. Armitage et al. have demonstrated optical addressing in FLCs [13]. Bistability makes this addressing scheme attractive since the write light need only be applied long enough to switch the FLC, after that the written image can be retained by the surface-stabilized FLC's intrinsic memory.

Photoaddressing spatial light modulators

Photosensors for SLMs have been fabricated from various materials, as summarized in Table I. Since the switching speeds of most of these devices is limited by the modulator to moderate values (milliseconds), the response-time demands placed upon the photosensors have been modest. The Hughes liquid crystal light valve, for example, uses a CdS photosensor [1, 2]. In this device a nematic liquid crystal is switched with a 30 msec cycle time. Other successful photoaddressed SLMs use crystalline silicon to switch nematic liquid crystals [3] and electro-optic crystals [5]. The response times are also on the order of milliseconds, although in the electro-optic crystal a cycle time of 0.5 milliseconds is predicted. Ashley and Davis have fabricated liquid crystal SLMs in which the photosensor is hydrogenated amorphous silicon [4]. Under low-intensity illumination, these devices exhibit a cycle time of 100 msec. With the advent of the FLC it becomes possible to fabricate SLMs which exhibit cycle times in the microsecond regime. This places substantially more severe demands upon the photosensor.

Table I
Comparison of Photosensors for Spatial Light Modulators (SLMs)

SLM	Photosensor	Contrast Ratio	Switching Voltage	Switching Speed	Resolution
Film	*AgBr*	10,000	--	--	> 1500 lp/mm
Hughes[1,2] LCLV	CdS	100	15 volt + optical power	30 msec	30 lp/mm
Hughes[3] LCLV	Si	100	15 volts + optical power	17 msec	15-100 lp/mm
Nematic[4]	a-Si H	100	12 volts	70 msec	> 35 lp/mm
Electro-optic[5] Crystals	Si	50% MTF	10^3 volts	15-2 msec	10 lp/mm
FLC[13]	BSO	5	10^3 volts	50 msec	3 lp/mm
SEED[19] (6 x 6 arrays)	GaAs/ GaAlAs	2	15 volts + optical bias and switching power	10 μsec	100 lp/mm
† FLC (128 x 128 arrays)	a-Si H	100-1000	15 volts	1 - 10 μsec	100 lp/mm

† proposed

Hydrogenated amorphous silicon (a-Si:H) may have the potential to fulfill the demanding requirements of FLC-based SLMs. Several aspects of a-Si:H make it particularly compatible with high performance SLMs.

(1) High photo-to-dark conductivity ratio. Under illumination of $1mW/cm^2$ this ratio is typically over three orders of magnitude [14]. The dark resistance across a film which is several microns thick is over $10^5\Omega/cm^2$.

(2) Large area coverage. As the material has been developed for use in flat panel solar cells, uniform layers of a sq. ft. may be deposited.

(3) Excellent spatial resolution. The diffusion length of holes is below 1 micron [15]. The requirement of space charge neutrality prohibits the photogenerated electron from migrating far from the photogenerated holes which, in turn, remain within 1 micron of the point of illumination because of their poor diffusion length. This localization of carriers corresponds to a spatial resolution of better than 100 lp/mm.

(4) Appropriate wavelength response. The optical absorption in a-Si:H rises sharply for wavelengths shorter than 700 nm, roughly corresponding to its band gap.

(5) Thin film. Because it is deposited as a thin film, transmission-mode as well as reflection-mode SLMs may be developed [16].

(6) Low temperature deposition. The deposition temperature is typically 250°C. Therefore low cost, temperature sensitive substrates, such as SnO_2 coated glass, may be utilized.

Configurations

Two basic circuit configurations for the photoaddressed SLM are shown in Figure 3a and 3b. In Figure 3a the FLC is depicted as a capacitor in parallel with a resistor, and the photosensor as a photodiode. In this configuration voltage is normally applied to reverse bias the photodiode. In the dark the photodiode passes very little current so the applied voltage is dropped across it. When the photodiode is illuminated it produces a current which charges the FLC and switches it to the ON state. The FLC may be switched OFF by reversing the polarity of the applied bias. The photodiode is then under forward bias and conducts so that the applied voltage is dropped across the FLC.

In Figure 3b the FLC is again depicted as a capacitor in parallel with a resistor, but here the photosensor is a photoconductor. In the dark the photoconductor is highly resistive so that the left side of the FLC is charged to V+ through the resistor R. When the photoconductor is illuminated its resistance is reduced, pulling the FLC to V-. When the FLC becomes sufficiently charged its state is switched ON. The FLC is switched OFF by terminating the illumination of the photoconductor, allowing the resistor R to once again pull the FLC to V+.

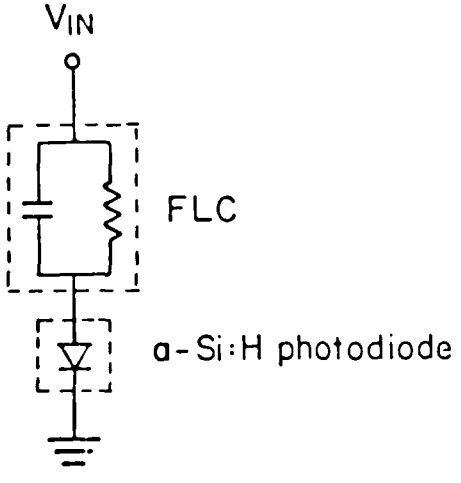


Figure 3a. Photodiode configuration.

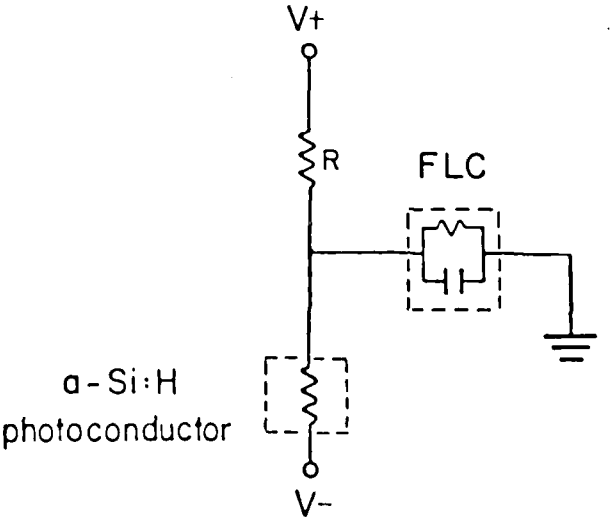


Figure 3b. Photoconductor configuration.

The photodiode permits the FLC to be switched OFF by applying a forward bias pulse and hence does not require a third element (such as a resistor) to have access to the FLC pixel. Therefore the entire FLC-photodiode array may be continuous, non-patterned layers. With the photoconductor the FLC may not be switched OFF simply by reversing the bias because, if the photoconductor is in the dark, the applied voltage will drop across *it* rather than the FLC. Therefore a discrete resistor must provide a voltage divider to each pixel, as shown in Figure 3b. This requires a photolithography step in the fabrication of the device to provide a resistor for each pixel. Alternative schemes include (i) forming a distributed resistor in a lateral device [16], (ii) switching OFF individual pixels on the entire array by illuminating the photoconductor while reversing the bias, and (iii) providing two photoconductor elements for each pixel, one pulling the FLC to V+ to switch it ON and one pulling the FLC to V- to switch it OFF.

Figures 4a and 4b show edge-on views of the photodiode and photoconductor reflection-mode devices. An a-Si:H p-i-n layer is deposited on SnO_2-coated glass to form the photodiode. A reflector separates the read beam from the write beam. If it is an insulating layer, the photosensor is capacitively coupled to the FLC. The FLC is sandwiched between the reflector and indium tin oxide (ITO) coated glass. The photoconductor is formed by providing an a-Si:H i-layer with two n-layers for ohmic contact. In Figure 4b a patterned device is shown in which a discreet resistor is connected to each pixel.

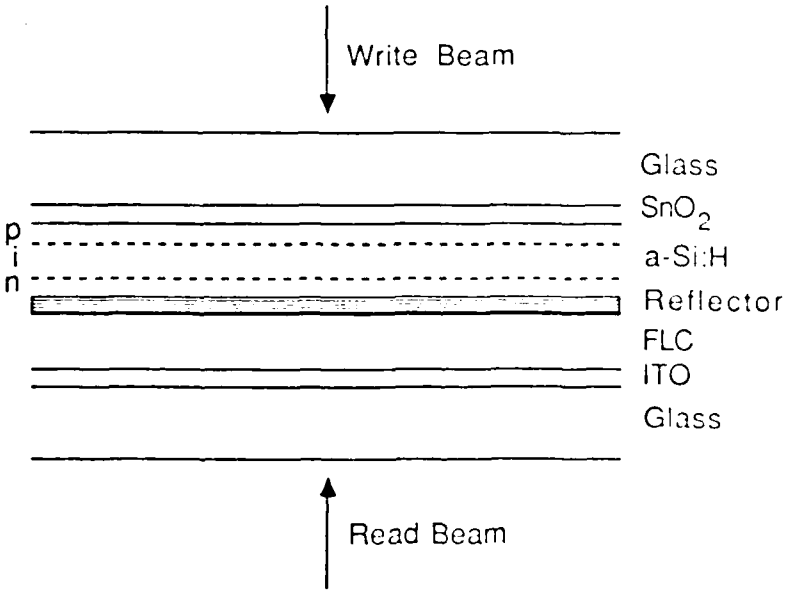


Figure 4a. Photodiode reflection-mode SLM.

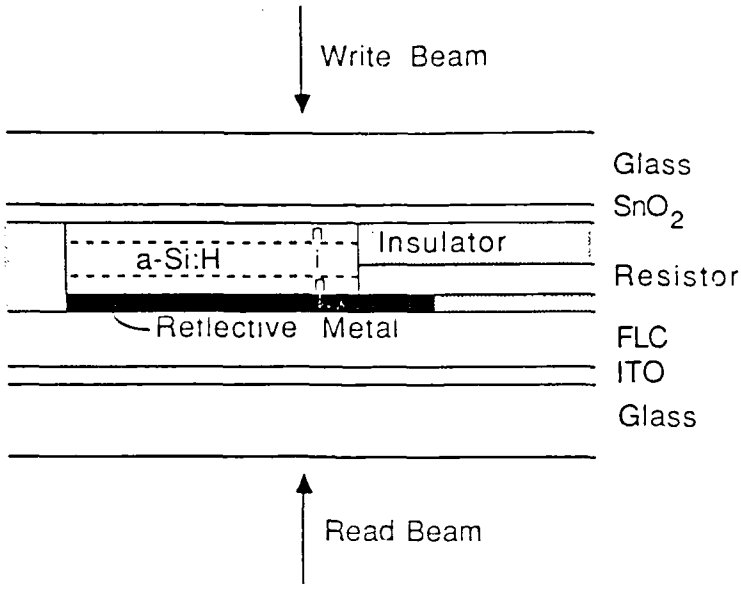


Figure 4b. Photoconductor reflection-mode SLM.

Response Time

For each configuration there are three factors which may limit the response of the device: the
required to charge the FLC, the intrinsic response time of the photosensor and the intrinsic switching
of the FLC. Since the switching time is inversely proportional to the polarization P, from Equation
desired FLC switching time determines P and hence the required charge. For example, the charge req
to reverse the state of the FLC in 1 μsec with 10 volts applied is 4×10^{-6} C/cm^2. For switching t
greater than 100 μsec the polarization charge becomes much less than the charge required by the geom
capacitance of the FLC and the photosensor. This capacitance is approximately 7nF/cm^2. The value c
parallel resistor is very large, so that the leakage current is negligible.

The time required to charge the FLC depends upon the illumination intensity and the characterist
sensor. For the reverse-biased photodiode, each absorbed photon produces one electron to cl
To generate the charge to switch the FLC in 10 μsec requires an illumination intensi
100mW/cm^2. A 30 msec cycle time, as would be required in a display application, v
μW/cm^2 illumination. The response time of the reverse biased a-Si:H photodiode itself
ant. Under a reverse bias of several volts this time is under 1 μsec [17], and ther
nse time of the devices.

each absorbed photon produces an electron which may traverse the c
Photoconductive gain results in much more charge per photon than in th
on intensity of 3mW/cm^2 should provide ample charge to switch the
time would require an intensity less than 1 μW/cm^2. The response ti
wever, is substantial. It can vary from microseconds to hundr
n conditions of the a-Si:H [18]. Unless the photoconduc
the limiting factor in the response of a photoaddressed SI

754 36

Conclusions

The combination of a ferroelectric liquid crystal and an a-Si:H photosensor provides the potential for a high speed photoaddressed spatial light modulator. The ferroelectric liquid crystal is capable of microsecond switching when it is provided with sufficient charge. An a-Si:H photodiode can provide this charge and response time, but it requires high illumination intensities. An a-Si:H photoconductor can provide the charge under lower illumination intensities, but the photoconductor response time may be a limitation.

Acknowledgements

The authors wish to thank Rodney Schmidt for helpful discussions. Support for this work was provided by the AFOSR under Contract No. 86-0189, IBM, AT&T, and Ball Aerospace industrial grants, and NSF Eng. No. 8451485.

References

[1] T. D. Beard, W. P. Beha, S-Y Wong, Appl. Phys. Lett. **22**, 90 (1973).

[2] J. Grinberg et al., Opt. Eng. **14**, 217 (1975).

[3] U. Efron et al., Proc. SPIE **388**, 75 (1983).

[4] P. R. Ashley, H. J. Davis, Appl. Opt. **26**, 241 (1987).

[5] D. Armitage, W. W. Anderson, T. J. Karr, IEEE J. Quant. Elec. **QE-21**, 1241 (1985).

[6] L. A. Pagano-Stauffer, K. M. Johnson, N. A. Clark, and M. A. Handschy, Proc. SPIE **684**, 88 (1986).

[7] N. A. Clark, M. A. Handschy, and S. T. Lagerwall, Mol. Cryst. Liq. Cryst. **94**, 213 (1983).

[8] M. A. Handschy and N. A. Clark, Ferroelectrics **59**, 69 (1984).

[9] M. A. Handschy and N. A. Clark, Ferroelectrics **59**, 25 (1984).

[10] Xue Jiu-Zhi, M. A. Handschy and N. A. Clark, Ferroelectrics **70-71** (1986).

[11] J. Wahl and S. C. Jain, Ferroelectrics **59**, 161 (1984).

[12] Chisso Corporation, Fine Chemicals Division, 2 Kamariya, Kanazawa-Ku, Yokohama 236, Japan.

[13] D. Armitage, J. Thackara, N. A. Clark, M. A. Handschy, Proc. SPIE **684**, 60 (1986).

[14] D. A. Anderson and W. E. Spear, Philos. Mag. **36**, 695 (1981).

[15] J. Dresner, D. J. Szostak and B. Goldstein, Appl. Phys. Lett. **38**, 998 (1981).

[16] P. R. Ashley and J. H. Davis, Proc. of the Army Sciences Conf. in West Point, June, 1986.

[17] T. Tiedje, C. R. Wronski, B. Abeles, and J. M. Cebulka, Solar Cells **2**, 301 (1980).

[18] G. Moddel, Ph.D. Thesis, Harvard University, 1981.

[19] D.A.B. Miller et al., 1986 annual meeting of the OSA.

Low Loss Polarization-Based Optical Logic Gates

W. Thomas Cathey
University of Colorado at Boulder
Department of Electrical Engineering and Computer Science
Boulder, Colorado, 80309-0425

Rodney A. Schmidt
University of Colorado at Denver
Department of Electrical Engineering and Computer Science
1100 14th Street, Denver, Colorado 80202

The gate design reported here is an outgrowth of a study of the use of optical computing for artificial intelligence.[1,2] The particular area chosen for study was mathematical resolution,[3,4] a method of reasoning using boolean algebra. In common with other non-arithmetic uses of optical computing, the operations involved in the process of mathematical resolution require the cascading of multiple levels of logic. For example, the basic operation in resolution, resolvent formation, requires on the order of $\log_2 n$ levels of logic to validate the resolvent of two clauses each containing n terms. Since in a real system, n might easily be 1024, and each level of logic might require two levels of optical gates, on the order of 20 gates might have to be cascaded in order to generate an output.

If logic levels are encoded as light polarization, it is possible to perform these operations without dissipative losses at each logic level. We are modeling the use of ferroelectric liquid crystals (FLC)[5-7] to do these operations. The gate makes use of the polarization rotation property of FLC devices to perform the boolean "and" or "or" functions for data represented using differing light polarizations. The architecture is suited for large systems of boolean variables in which data must be processed in parallel. Efficient use is made of optical components in that only the liquid crystal devices and photodetectors require an internal array structure. The remaining devices are image preserving, and only one optical element is required at each step, regardless of the size of the data array being processed. The array structure of the photodetectors and FLC elements requires only local coupling, with each FLC connected to an immediately adjacent detector. This would facilitate the layout of entire matrices of gates. The parallel nature of the process, combined with the high speed of ferroelectric devices, allows the system to operate at rates comparable to electronic processors. The optical gate presented is the kernel of a larger system which includes data storage and control. Because of the general applicability of the gate design, it will be presented separately here. The larger system will be presented at a later date.

The fundamental properties of a cascadable gate are well known. The inputs and outputs of the gate must be spatially equivalent (same area, same collimation), optically equivalent (same frequency, same coherence, same polarization) and either the gate must regenerate amplitude, or losses must be small enough that a reasonable number of gates can be cascaded before losses become unacceptable. The present design satisfies all these criteria.

Input to the gate is in the form of linearly polarized light. For a two-input gate, the inputs are presented spatially adjacent, as either horizontally or vertically polarized light. The pair of inputs passes through a birefringent crystal. The crystal separates the horizontal and vertical components of each input (in the absence of polarization noise, each input will contain only one polarization). The horizontal components of both inputs impinge on a single semi-transparent photodetector, which intercepts a small fraction of the

incident light. The resistance change of the photoconductive detector is used to change the voltage applied to a single ferroelectric liquid crystal located in the path of both of the vertically polarized input components. If no light impinges on the photodetector, the FLC does not rotate light passing through it. If light hits the photodetector, the FLC performs a 90 degree rotation on light passing through it. Since only vertically polarized light passes through the FLC, if there is no horizontal light on the photodetector, the gate output is vertically polarized. If either input is horizontally polarized, the FLC will ensure that any vertical component of the other input will be shifted to horizontal. In addition, almost all of the horizontal input is passed through the system, since only a small portion is used to drive the photodetector.

The output of the gate occupies four times the spatial area of the input, and potentially is distributed in different ways, depending on the input combinations. The size is reduced to match the original by a converging-diverging lens pair which effects a 2:1 scale change in both dimensions. In an array of gates, only one lens pair is needed for the entire system, so this does not add excessive complexity. The spatial distribution of the light within the output pattern is not significant because subsequent gates are uniformly sensitive over their entire input window.

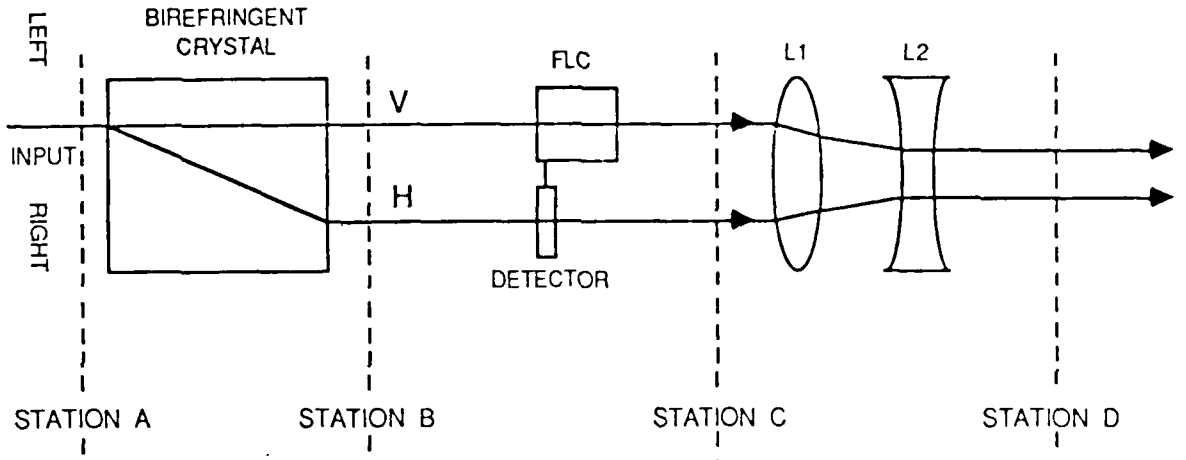
A diagram of the gate is shown in Figure 1. The input light enter at station A. A birefringent crystal separates the light into horizontally and vertically polarized components as shown at station B. An FLC-detector pair transmits the horizontal light, and may or may not rotate the vertical light to horizontal. This is shown at station C. A converging-diverging lens pair at station C shrinks the output pattern to be the same size as the input. The output at station D is ready for deflection into further gates.

The gate is essentially nonabsorptive, with the only intentional light absorption occuring in the photodetector. This is deliberately inefficient, with the majority of the input light passing through to the output. The gate does require electric power to drive the FLC, but the FLC is very high impedance, and the switching energy is very low, on the order of 0.8 picojoules for a typical gate. The electrical circuit of the gate is shown in Figure 2.

Our mathematical resolution system will use monolithic arrays of optical gates. Each gate array will consist of a crystal with a matrix of FLC's and detectors deposited on it, and a single converging-diverging lens pair for the entire array. Our present experimental results are drawn from a proof-of-concept model fabricated on an optical bench. This model is shown in Figure 3.

Our present work involves the detailed simulation of the physics of the monolithic gate array. This will then be applied to a simulation of the larger artificial intelligence problem while fabrication of a sample array is begun.

A) TOP VIEW OF CIRCUIT LAYOUT



B) AXIAL VIEW OF SIGNAL PATH

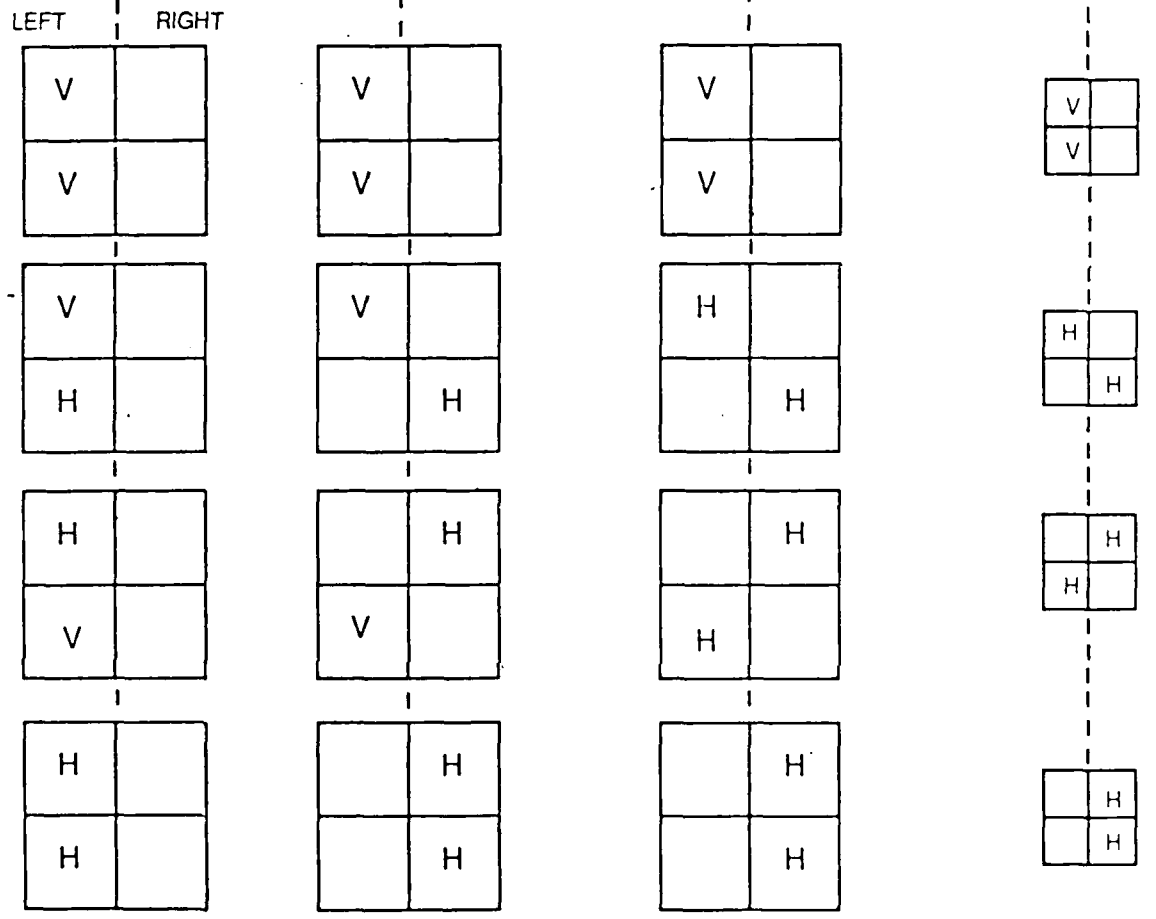


FIGURE 1

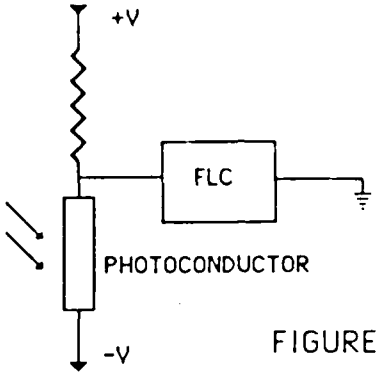


FIGURE 2

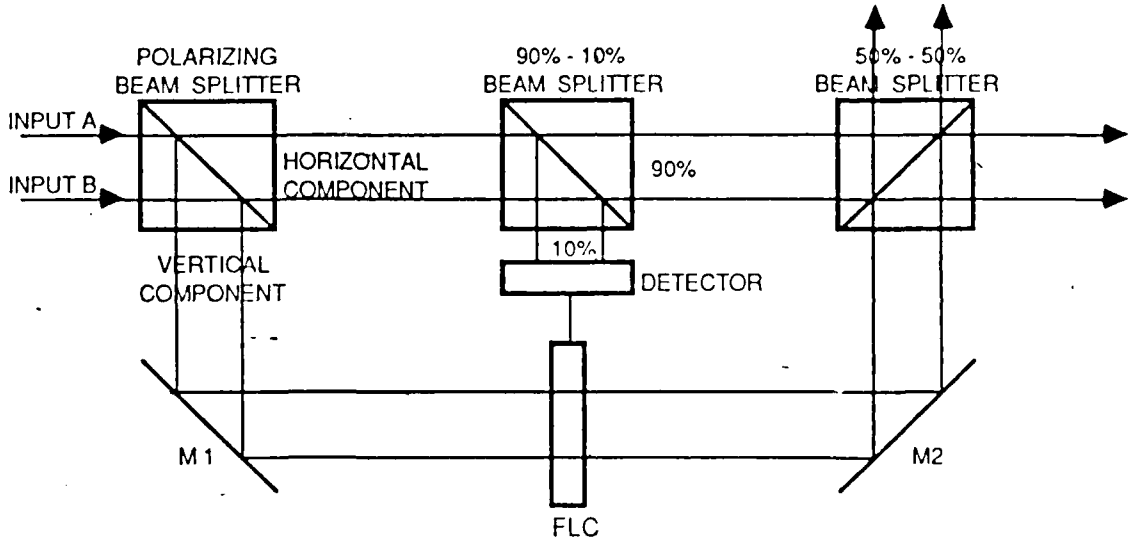


FIGURE 3

References

1. Schmidt, R. A. and W. T. Cathey, "Optical Implementations of Mathematical Resolution", Applied Optics, March 1987 (to be published).

2. Schmidt, R. A. and W. T. Cathey, "Optical Representations for Artificial Intelligence Problems", SPIE Proceedings, Vol. 625, p. 226, 1986.

3. Nilsson, N. , Principles of Artificial Intelligence, Tioga, 1980.

4. Winston, P. H., Artificial Intelligence, Addison Wesley, 1984.

5. Clark, N. A. and S. T. Lagerwall, "Submicrosecond Bistable Electro-Optic Switching in Liquid Crystals", Applied Physics Letter, 36(11), June 1980.

6. Clark, N. A. and S. T. Lagerwall, "Physics of Ferroelectric Fluids: the Discovery of a High-Speed Electro-Optic Switching Process in Liquid Crystals", Recent Developments in Condensed Matter Physics, Vol. 4, 1981.

7. Handschy, M. A. and N. A. Clark, " Stroboscopic Microscopy of Fast Electro Optic Switching in Ferroelectric Smectic C Liquid Crystals", Applied Physics Letter, 41(1), July 1982

Polarization-based optical parallel logic gate utilizing ferroelectric liquid crystals

M. A. Handschy, K. M. Johnson, and W. T. Cathey

Center for Optoelectronic Computing Systems

University of Colorado

Boulder, CO 80309-0425

Abstract

We describe parallel optical XOR and XNOR logic gates implemented with ferroelectric liquid crystal (FLC) electrooptic elements to yield naturally a logic where the binary gate outputs are indicated by two orthogonal polarizations of transmitted light. These gates absorb no power from the incident light beam, and hence can be cascaded without need to regenerate NOT inputs. The FLC elements also confer the advantages of low-voltage, low power, submicrosecond switching and intrinsic two-state memory.

I. Introduction

Optical computing systems offer increased information processing rate by facilitating parallel computing architectures. Previous experience with electronic computers indicates that desired accuracy can be achieved only with digital computation. Since the simplest digital arithmetic is binary, most recent work on optical computing has focused on the construction of binary optical logic gates. While most practical implementations of optical logic gates represent the binary logic states by two light intensity levels (see a recent review by Sawchuck and Strand [1]), another perhaps more natural implementation represents the two states with two orthogonal polarizations of light. An early description of a polarization based logic was given by Watrasiewicz [2], using dichroic elements that could be switched to absorb either of two orthogonal incident polarizations. Lohmann [3] has recently pointed out that polarization based logic implemented with nonabsorptive elements has the advantage that light need not be lost in the logical operation of inversion, preventing unwanted dissipation of optical power, and allowing the cascading of many gates. He has also described an implementation of such a logic using a nonlinear element that has different transmission and reflection vs. intensity functions for different polarizations of incident light.

Any optical element, that in one state converts the polarization of incident light to its orthogonal complement while in another state transmits incident light with its polarization unchanged, could also be used to implement a polarization based logic that has these advantages. Common examples of such devices include variable retarders, such as those using the Pockels effect or the variable birefringence mode in nematic liquid crystals. The now common twisted nematic liquid crystal device was used by Tavetkov et

al [4] as a nonabsorptive polarization rotator in the earliest implementation known to the present authors of a polarization based logic of any kind.

We describe here a new implementation, in which the optical element is a ferroelectric liquid crystal device that functions as a half-wave plate whose axis can be electrically toggled between two orientations that make a 45° angle to each other. These elements have extremely useful operating characteristics for optical parallel processing, including fast response time (submicrosecond), low power, low voltage switching (tens of volts), and bistability [5]. FLC elements have already been used in an intensity based logic scheme where their high contrast (up to 1500) has been exploited to advantage [6]. Here, we show how they can be used in polarization-based logic to give especially simple implementations of the XOR and XNOR logical operations.

II. FLC ELECTROOPTICS

Ferroelectric liquid crystals possess properties especially attractive for optical logic applications when used in the so-called surface-stabilized geometry, which has been described extensively elsewhere [7,8,9]. Briefly, a slab of essentially optically uniaxial FLC is disposed between two closely spaced glass plates, coated on their inner surfaces with a transparent electrical conductor. Voltages of opposite sign applied to the plates select between two optic axis orientations, both parallel to the plates, but differing in direction by an angle $2\psi_a$. The "tilt angle" ψ_a, a material property of the FLC, is typically close to 22.5° over large temperature ranges, allowing the optic axis to be electrically rotated through approximately 45°. If the thickness d of the FLC layer is chosen so that $\Delta n d = \lambda/2$, where Δn is the FLC's birefringence (typically 0.1 - 0.2) and λ is the vacuum wavelength of the incident light, the FLC becomes a half wave plate.

If the polarization of normally incident light is chosen either parallel or perpendicular to one of the voltage selected optic axis states, it will be transmitted through the FLC unaffected. The optic axis state selected by the opposite applied voltage is then 45° to the incident polarization, so that both the ordinary and extraordinary modes will be excited. For correct FLC element thickness d, a total phase shift of π will accumulate between these two modes, and the incident light's polarization will be rotated by 90°. Fig 1(a) schematically illustrates this action of the FLC electrooptic element.

Beside the previously mentioned switching speed, the surface stabilized FLC geometry offers another feature useful in optical logic systems: bistability. After either applied voltage brings the optic axis to one of its preferred orientations, that voltage may be removed without the optic axis returning to its previous state. This allows a two dimensional array of FLC elements to be matrix addressed by row and column electrodes. A practical scheme for accomplishing this has been demonstrated by Wahl et al. [10], who achieved 1000:1 multiplexing. Thus, a large number of FLC elements (1000 × 1000 = 10^6) can be simply fabricated on a single substrate, and driven with an economical number of electrical connections. Optical addressing has also been demonstrated in FLCs [11]. Bistability makes this addressing scheme attractive since the write light need only be applied long enough to switch the FLC; after that the written image can be retained by the surface stabilized FLC's intrinsic memory.

III. FERROELECTRIC LIQUID CRYSTAL LOGIC GATE

The XOR (AB'+A'B) and XNOR (AB+A'B') Boolean functions are difficult to implement optically using bright and dark logic because the light that is irretrievable lost when creating not A (A') and not B (B') must be regene

rated. The simple optical gate shown in Fig. 1(b) using two FLC arrays and an optical controller takes advantage of the lossless properties of polarization logic to realize these functions. Vertically polarized laser light illuminates FLC array A, which is a programmable matrix made up of transparent pixel elements. The "nonswitched" pixels have their optic axis vertical, and transmit the incident light with unchanged polarization, while the "switched" pixels, with their optic axis at 45° to the vertical, rotate the polarization of the incident light to horizontal. Now a pattern made up of horizontally and vertically polarized light illuminates FLC array B. If either vertically or horizontally polarized light illuminates a switched pixel in FLC B, the polarization is rotated by 90°, vertical rotates to horizontal and horizontal rotates to vertical. Light incident on a non-switched pixel is transmitted with unchanged polarization. The truth table of Fig. 2 summarizes the logical function. The controller passes the light unchanged, and an analyzer at the output provides visual inspection of the XOR function. To realize the XNOR, the FLC optical controller, which is a uniform electrically switchable FLC half wave plate, is switched, and the two output polarizations are interchanged. The truth table for XNOR is also shown in Fig. 2.

IV. PERFORMANCE AND LIMITATIONS

FLC devices that function exactly as described in Section II result in logic gates with exactly orthogonal output polarization states. Several deviations from the ideal can compromise this performance, reducing the quantity that in a usual electronic logic would be called noise immunity. In the unswitched state, output light of the undesired polarization component results only from defects in the FLC's smectic layer orientation; by appropriate alignment techniques these defects can be minimized and the

intensity of the unwanted polarization component made quite small as shown by the large extinction ratios obtainable with these devices between crossed polarizers [6]. In the switched state, output light of the undesired polarization results from the FLC element's deviation from an ideal half wave plate; the ratio of the intensity of the unwanted polarization component to the incident intensity is $1 - \sin^2(4\psi_A)\sin^2(\pi\Delta nd/\lambda)$. This allows either ψ_A or d singly to deviate from their ideal values (22.5° and $\lambda/(2\Delta n)$ respectively) by up to 8% without the unwanted intensity increasing above 1% of the incident intensity (100:1 noise immunity).

The switching speed of the FLC element for a given applied electric field strength E is largely determined by the FLC material's ferroelectric polarization P and viscosity η, through the relation $\tau = \eta/(PE)$. Optical 10 - 90% times are usually about 1.8τ [12]. The viscosity of typical materials can be determined from the above relation when P is known. For the high temperature material HOBACPC, this relation gives $\eta \approx 3$ cP at the lowest temperature in its smectic C* phase [13]. For a typical room temperature material, CS 1014, one can infer $\eta \approx 50$ cP from the manufacturer's data [14]. Modest improvements over currently available polarizations should yield P's of 1 $\mu C/cm^2$. With this polarization, applied electric fields of 100 V/μm would give switching times of about 5 ns at elevated temperature and 100 ns at room temperature. However, the speed of an FLC array is more likely to be limited by its maximum allowable power dissipation than by the switching time of its FLC material. Switching a unit area of FLC by reversing an applied voltage V dissipates an energy 2PV through the reversal of the polarization. If this reversal is repeated as frequently as possible (i.e. once every 1.8τ), the power dissipated is $2PV/(1.8\tau) \approx \eta d/\tau^2$, where d is the FLC thickness ($E = V/d$). Thus, for a given maximum allowable power dissipation W, the shortest achievable characteristic time is given by $\tau = [\eta d/W]^{\frac{1}{2}}$. For operation at room

temperature of an FLC device of d = 1 μm with a dissipation of W = 100 mW/cm². this relation gives a minimum switching time of 13 μs. at an elevated temperature with W = 1 W cm², the minimum switching time is reduced to 550 ns

V CONCLUSIONS

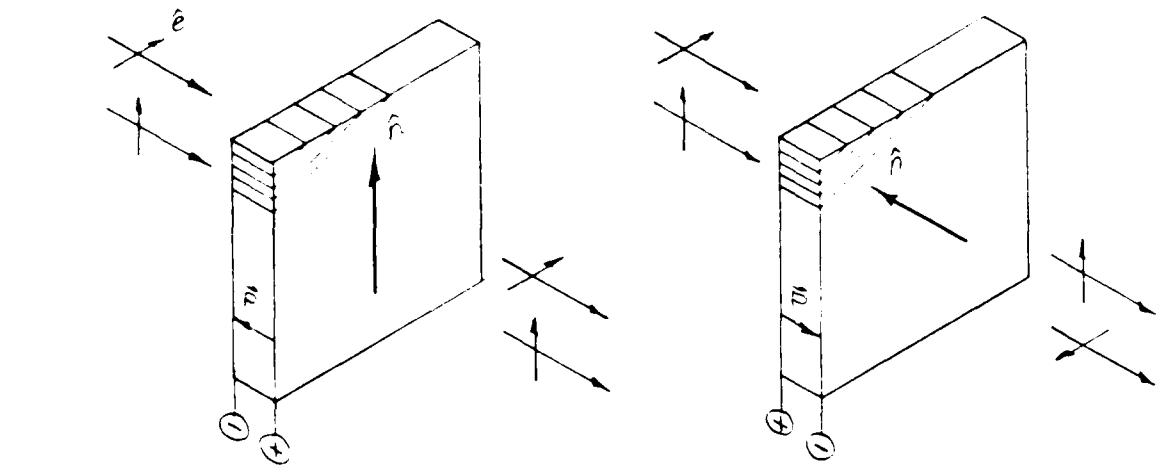
The FLC optical logic gate provides high switching speed with low power dissipation, high noise immunity, and low threshold switching voltage at a low cost. This device is of particular interest in optical parallel processing applications because it is based on polarization logic, which has several advantages over bright/dark logic: the output is naturally binary having only the two complementary polarization states, the XOR and XNOR functions are easily implemented with only two non emitting FLC arrays; and most importantly FLC logic gates can easily be cascaded since no light is lost in performing each logical operation. At the end of N operations, a single analyzer can be used to yield an opaque and transparent image for visual inspection. By comparison, the typical bright/dark logic scheme would have an analyzer after every input device, and would lose on average 50% of the available light per operation, limiting the number of gates that could be cascaded without unacceptable light loss.

ACKNOWLEDGMENTS

The authors were supported by grants from IBM (MAH and KMJ), from the NSF, (DMR-8451484) AT&T Informations Systems Division and Ball Aerospace (KMJ); and the U. S. Air Force AFOSR 86 0189 (WIC).

References

1. Alexander A. Sawchuck and Timothy C. Strand, Proc. IEEE 72, 758 (1984).
2. B. H. Watraslewicz, Optics Laser Technol. 7, 213 (1975).
3. Adolf. W. Lohmann, Appl. Opt. 25, 1594 (1986).
4. V. A. Tsvetkov, N. A. Morozov, and M. I. Ellinson, Sov. J. Quant Electron. 4, 989 (1975).
5. N. A. Clark and S. T. Lagerwall, Appl. Phys. Lett. 36, 899 (1980).
6. L. A. Pagano Stauffer, K. M. Johnson, N. A. Clark, and M. A. Handschy, Liquid Crystals and Spatial Light Modulator Materials, William A. Penn, Ed., Proc. SPIE 684, 88 (1986).
7. N. A. Clark, M. A. Handschy, and S. T. Lagerwall, Mol. Cryst. Liq. Cryst 94, 213 (1983).
8. M. A. Handschy and N. A. Clark, Ferroelectrics 59, 69 (1984).
9. Noel A. Clark and Sven T. Lagerwall, Ferroelectrics 59, 25 (1984).
10. J. Wahl, T. Matuszczyk, and S. T. Lagerwall, to appear Mol. Cryst. Liq. Cryst. (1986).
11. D. Armitage, J. Thackara, N. A. Clark, and M. A. Handschy, Liquid Crystals and Spatial Light Modulator Materials, William A. Penn, Ed., Proc. SPIE 684, 60 (1986).
12. Xue Jiu zhi, M. A. Handschy, and N. A. Clark, to appear Ferroelectrics 70 71, (1986).
13. J. Wahl and S. C. Jain, Ferroelectrics 59, 161 (1984).
14. Chisso Corporation, Fine Chemicals Division, 2 Kamariya, Kanazawa ku, Yokohama 236, Japan.



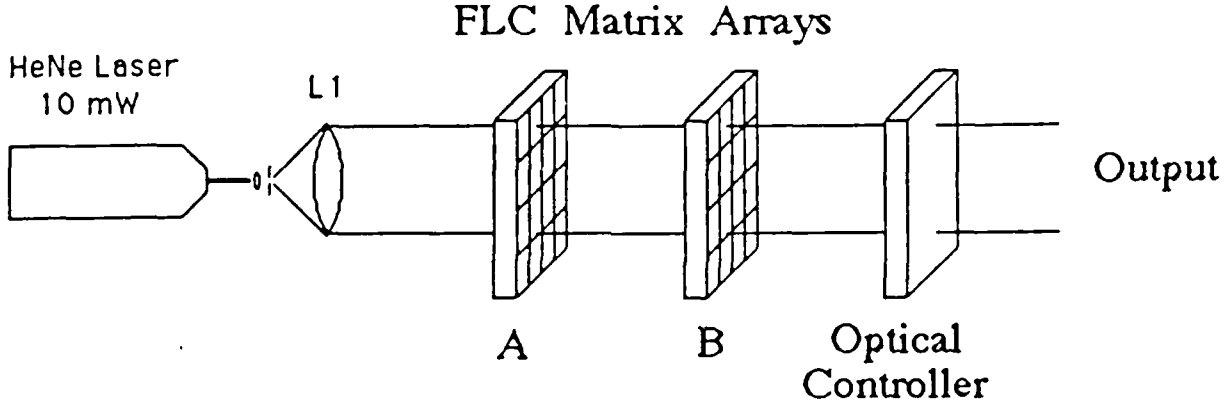


Figure 1. (a) Electrooptic action of ferroelectric liquid crystal (FLC) element. The FLC is in the form of a thin slab between transparent conductive electrode plates (not shown). The liquid crystal's smectic layers make an angle ψ_0 to the horizontal. Also shown is the optic axis direction (director) $\hat{n}$, the ferroelectric polarization $\vec{P}$, and normally incident light rays with polarizations $\hat{e}$ vertical and horizontal. In the state shown to the left with $\hat{n}$ vertical, selected by "positive" applied voltage, the incident light is transmitted with polarization unchanged; in the state shown to the right with $\hat{n}$ at $2\psi_0 \approx 45°$ to vertical, selected by "negative" applied voltage, the polarization of either incident ray is rotated by 90°. (b) Two arrays of FLC elements arranged to make XOR and XNOR optical logic gates. The controller is a large, single element.

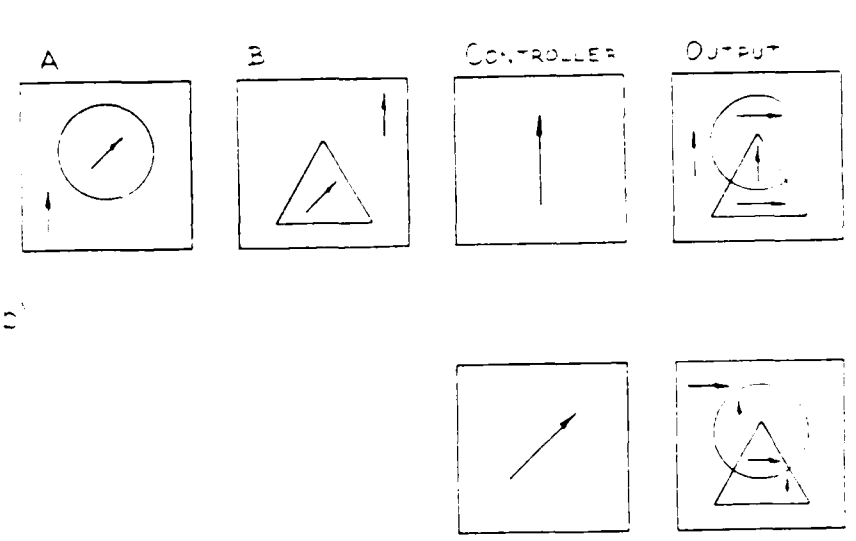


Figure 2. Truth table for XOR and XNOR functions. For inputs A and B and the optical controller, the arrows represent the direction of the FLC optic axis. For the output, the arrows represent the polarization direction. (a) XOR. (b) Switching the controller reverses the output polarizations, yielding the XNOR function.

END

8-87

DTIC